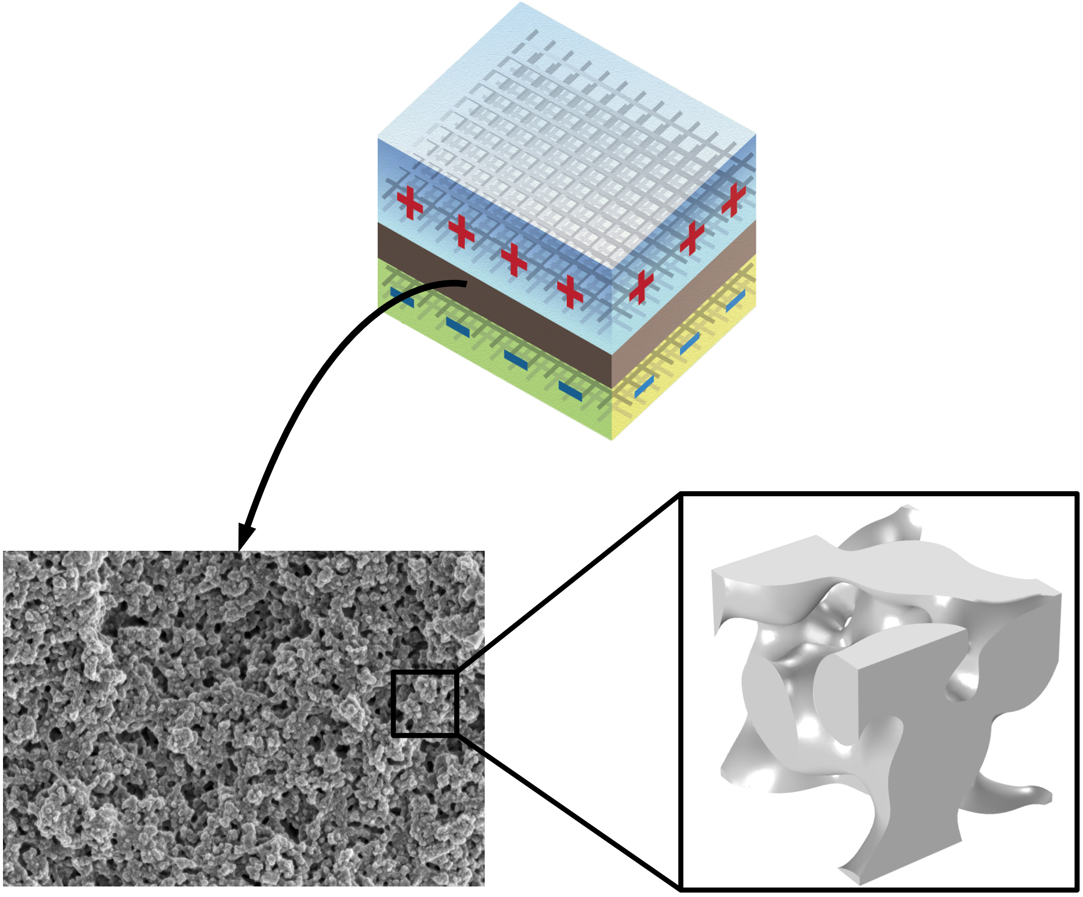
CHALMERS

UNIVERSITY OF TECHNOLOGY



Modeling and Finite Element Simulation of the Bifunctional Performance of a Microporous Structural Battery Electrolyte

Master's thesis in Applied Mechanics

VINH TU

Department of Industrial and Materials Science
Chalmers University of Technology
Gothenburg, Sweden 2019

Master's thesis

Modeling and Finite Element Simulation of the Bifunctional Performance of a Microporous Structural Battery Electrolyte

VINH TU

Department of Industrial and Materials Science
Division of Material and Computational Mechanics
Chalmers University of Technology
Gothenburg, Sweden 2019

Modeling and Finite Element Simulation of the Bifunctional Performance of a Microporous Structural Battery Electrolyte

VINH TU

Supervisors:	Associate Professor Ralf Jänicke, Industrial and Materials Science Professor Leif Asp, Industrial and Materials Science Professor Fredrik Larsson, Industrial and Materials Science Professor Kenneth Runesson, Industrial and Materials Science
Examiner:	Associate Professor Ralf Jänicke, Industrial and Materials Science

Master's Thesis
Department of Industrial and Materials Science
Division of Material and Computational Mechanics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Laminated structural battery [5] at the top, SEM image of SBE [4] to the bottom left, and artificial SBE microstructure to the bottom right.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2019

Modeling and Finite Element Simulation of the Bifunctional Performance of a Microporous Structural Battery Electrolyte
Master's thesis in Applied Mechanics
VINH TU
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

The structural battery composite is an innovative solution for a light-weight storage of electrical energy. It is multifunctional since it carries mechanical loads and stores electrical energy simultaneously. Such multifunctional materials will become important for e.g. the electrification of vehicles since large weight-reductions can be gained. This innovation will contribute greatly to realizing the vision of a carbon-neutral circular economy. Due to the infancy of this technology, the structural batteries still need to be developed further.

The main purpose of the project is to model and simulate the diffusive transport of lithium ions through the structural battery electrolyte (SBE) between the structural battery electrodes in a laminar setup, and to simulate the SBE's mechanical behaviour. The SBE is a microporous polymer matrix filled with a liquid electrolyte. By generating an artificial SBE microstructure and performing some virtual material testing, it will become possible to evaluate the effective multifunctional performance of the SBE for varying pore sizes.

The artificial SBE microstructure is generated by manipulating the stationary heat equation. It is possible to choose heat sources and heat sinks in a clever fashion in order to obtain the desired shape for the artificial microstructure. The choice of heat sources and heat sinks is based on a periodic Voronoi tessellation that is embedded in a solid unit cube. By letting the Voronoi edges be heat sources, and the Voronoi seeds be heat sinks, an artificial microstructure which is microporous and bicontinuous is obtained after some modification and post-processing of the temperature field.

The multifunctional performance of the SBE is evaluated by applying the theory of computational homogenization on the artificial SBE microstructure which serves as a statistical volume element with the ability to almost fully characterize the material's heterogeneities. In particular, the weakly periodic boundary conditions are used. The results from the virtual material testing indicate that the lower bound of the stiffness increases for increasing volume fraction of polymer matrix, while the upper bound of the stiffness and the ionic conductivity decreases. Furthermore, the effective diffusivity seems to scale linearly with the volume fraction while the bounds of the effective stiffness seem to scale non-linearly.

Although the goal of the project is to mimic the SBE in the structural battery, the end result is a quite general recipe on artificial microstructure generation. Nevertheless, this thesis paves the way for more rigorous artificial SBE microstructure generation in the future.

Keywords: Structural battery, porous media, artificial microstructure, statistical volume element, computational homogenization, virtual material testing

Preface and acknowledgements

This project is the final piece of the master's programme Applied Mechanics at Chalmers University of Technology. It was carried out during the spring semester of 2019 at the division of Material and Computational Mechanics, department of Industrial and Materials Science.

First of all, I want to thank my main supervisor Ralf Jänicke for his excellent guidance and active support throughout the whole project. I would also like to thank my co-supervisors Leif Asp, Fredrik Larsson and Kenneth Runesson for engaging in invaluable discussions pertaining to my Master's thesis project. Many thanks to David Carlstedt for helping me understand some of the finer details of the structural battery. Lastly, I want to thank Monica for her love, support, and for accompanying me at school even during weekends and holidays.

Vinh Tu, Gothenburg, May 2019

Contents

1 Introduction

Carbon fibers are traditionally combined with a plastic resin to form a carbon fiber reinforced polymer (CFRP) which is a high strength-to-weight ratio material when loaded in a favorable direction. However, as of late it has been shown that carbon fibers are high capacity Li-ion battery negative electrodes. This means that carbon fibers can carry load, and at the same time intercalate lithium ions in a similar way as the negative electrode in lithium ion batteries [1, 2]. This gives rise to a new multifunctional material, namely the structural battery composite.

The structural battery composite is an innovative light-weight multifunctional material since it has both structural and electrical energy storage functionalities [2]. A more holistic approach where two different needs are addressed by a multifunctional material instead of two separate subsystems results in huge savings that otherwise would be limited when considering the subsystems individually [3]. For example, the multifunctional structural battery composite can contribute greatly to the electrification of vehicles where large weight-reductions can be gained [4]. In the case of a traditional electric car, the vehicle frame offers no extra benefit other than supporting the mechanical loads, and the heavy battery pack offers no extra benefit other than storing the electrical energy. By utilizing the structural battery composite in the vehicle frame, the electric car could potentially exclude the heavy battery pack, and thus reducing the mass significantly. The bottom line is that this innovation enhances the system performance in various applications, and addresses the demand for more efficient and sustainable systems. However, due to the infancy of this technology, the structural battery composite still need to be investigated further.

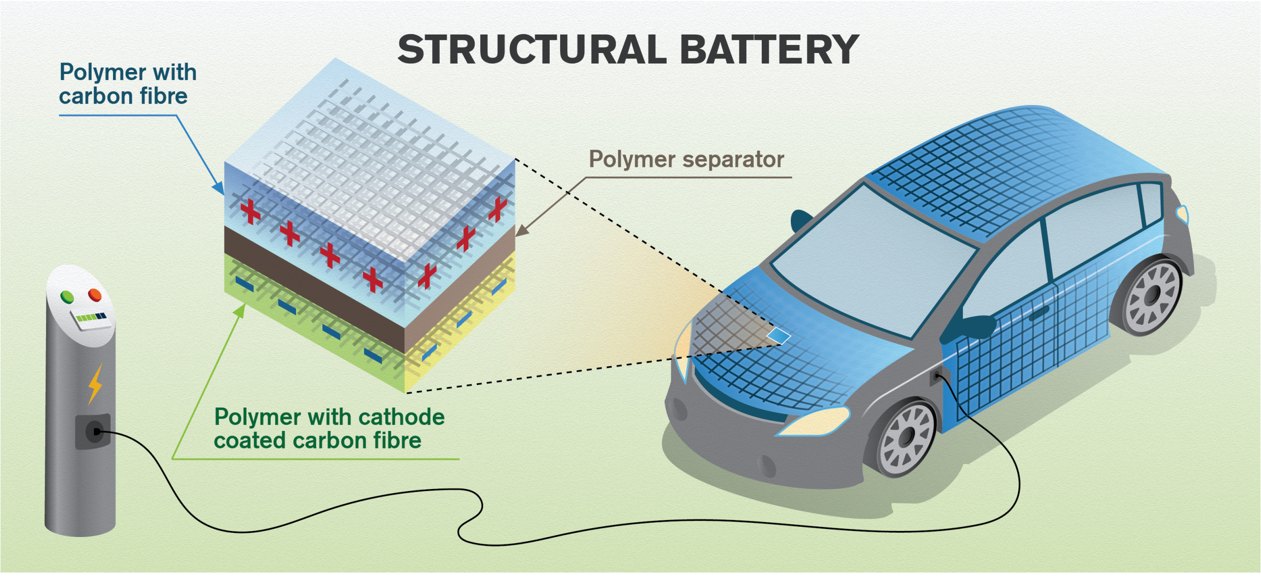

STRUCTURAL BATTERY
Polymer with carbon fibre
Polymer separator
Polymer with cathode coated carbon fibre

Figure 1.1: Futuristic concept of utilizing structural battery in an electric car. Reproduced with permission by Leif Asp and Yen Strandqvist [5].

1.1 Structural battery architecture

The structural battery can be realized based on the laminated battery concept or the 3D battery concept. It is the laminated architecture that is of interest in this project. The laminated battery was first introduced by Wetzel and his team at the United States Army Research Laboratory [6]. This battery consists of several layers that have different functions. One of the layers is called the negative electrode lamina, and the opposite layer is called the positive electrode lamina. In-between these two layers is a polymeric separator layer. Lastly, the current collectors correspond to the uppermost and lowermost layers. The negative electrode lamina consists of carbon fibers that are embedded in structural battery electrolyte (SBE), and it is this lamina that utilizes the carbon fiber's ability to act as high capacity Li-ion battery negative electrodes. The positive electrode lamina is exactly the same, except that the carbon fibers are covered in a lithium-metal-oxide doped coating (e.g. $LiFePO_4$) in order to act as the Li-ion battery positive electrodes. The SBE is a microporous polymer matrix that is submerged in a liquid electrolyte. Furthermore, the SBE is bicontinuous, i.e. both the solid polymer foam and the pore space are continuous. Its function is to be electrically insulating but ionically conductive while being able to carry mechanical loads. It is this polymer matrix that will be modeled and simulated in the project. The separator layer between the electrodes is an extra safety layer to prevent electrical short circuiting in case the electrodes come into contact. This separator layer can be manufactured in the same material as the SBE. As in conventional batteries, the ion transport occurs through the layers between the electrodes while the electrons transport to the current collectors via the carbon fibers.

The 3D battery was first presented by Carlson [7], and Asp et al. [8]. It is different from the laminated architecture since it is not built up by several layers, instead this concept uses individual carbon fibers as battery electrodes. The 3D battery consists of carbon fibers that are embedded in a polymer matrix similar to traditional CFRP, but the carbon fibers are more spread out. In this case, the carbon fiber is still the negative electrode, but the polymer matrix is the positive electrode since it is doped with lithium-metal-oxide based particles. Each carbon fiber is covered in a thin polymer coating which acts as the separator and electrolyte. Due to this, each carbon fiber embedded in the doped polymer matrix corresponds to a battery cell. Furthermore, unlike the laminated architecture, the ion transport occurs in a radial direction out of each carbon fiber during discharge.

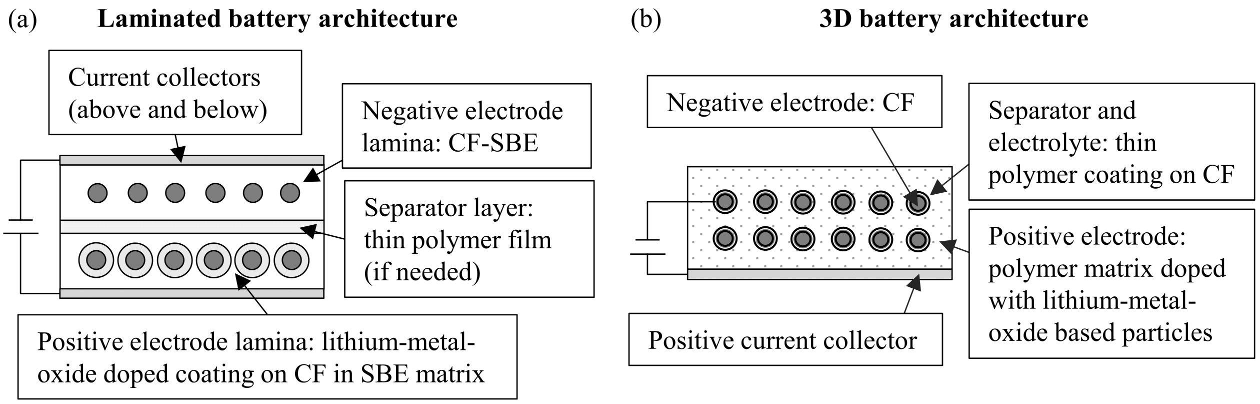

(a)
Laminated battery architecture
Current collectors (above and below)
Negative electrode lamina: CF-SBE
Separator layer: thin polymer film (if needed)
Positive electrode lamina: lithium-metal-oxide doped coating on CF in SBE matrix
(b)
3D battery architecture
Negative electrode: CF
Separator and electrolyte: thin polymer coating on CF
Positive electrode: polymer matrix doped with lithium-metal-oxide based particles
Positive current collector

Figure 1.2: Schematic overview of the (a) Laminated architecture and the (b) 3D architecture of structural batteries. Reproduced with permission by David Carlstedt [9].

1.2 Problem statement and approach

The main purpose of the project is to model and simulate the diffusive transport of Lithium ions through the SBE between the structural battery electrodes in a laminar setup. As mentioned above, the polymer matrix is microporous and submerged in a liquid electrolyte. It turns out that the transport of the lithium ions and the mechanical behaviour are dictated by the size of the pores in the polymer matrix. By performing virtual material testing on artificial microstructures of the SBE, it will become possible to evaluate the effective multifunctional performance of the SBE for varying pore sizes.

In order to carry out the project, several sub-goals need to be achieved:

- Generation of artificial microstructures representing the microporous polymer matrix, i.e. the SBE. The generated artificial microstructure must be a statistical volume element (SVE).
- Modeling and finite element simulation based on stationary diffusion and linear elasticity.
- Computation of the effective multifunctional performance, i.e. the ionic conductivity and mechanical load transfer of the polymer matrix for varying pore sizes.

1.3 Project limitations

In order to reduce the complexity of the problem and to fit the project into a Master's thesis course, several limitations are made:

- Charging and discharging of the structural battery is a truly transient problem, but for simplicity, only the stationary lithium ion transport problem will be investigated.
- The fully coupled electrochemical equations are simplified into a single chemical diffusion problem.
- Although this project focuses on the linear elasticity and the diffusion problem, they are not solved in a coupled way. They are decoupled for simplicity.
- For this project, the carbon fibers will not be modeled inside the SBE. Thus, there is no interaction between the SBE and the fibers. In short, this is an analysis of an isolated constituent, i.e. the SBE, of the structural battery. However, this analysis does represent the separator layer well since there are no CF-electrodes in it.
- It is known that during charge and discharge cycles, the structural battery generates heat due to ohmic heating. This is concerning due to the fact that polymers in general are vulnerable to heat, i.e. a large portion of the mechanical properties can be lost if high temperatures are obtained. Furthermore, heat will affect the volume and elastic properties of the constituents, which induces internal stresses. However, all thermal effects are neglected in this project.

2 Two-scale modeling approach

Computational homogenization is a well-established method that includes the effects of micro-heterogeneity in material substructures when dealing with constitutive modeling. From a computational standpoint, it is much more feasible to introduce a separation of the macro- and sub-scales, rather than trying to directly resolve a model which captures both scales. The theory of computational homogenization is often used in conjunction with the FE^2-method where one couples two different geometric scales and solves the problem in a fully nested fashion. The aim is usually to obtain effective properties for the macro-scale via homogenization of the sub-scale problem, where the sub-scale problem is defined by a Representative Volume Element (RVE) that can fully characterize the heterogeneity in the material substructures [12]. In such a case, every macro-scale point $\bar{\boldsymbol{x}} \in \Omega$ corresponds to an RVE in the sub-scale domain $\Omega_\square$. However, in this project the focus will be on the homogenization of the sub-scale output in order to obtain the effective properties. The homogenized effective properties are never re-incorporated into a macro-scale problem, i.e. no FE^2-approach will be employed as one would do in a true multi-scale analysis. The theory of computational homogenization is presented in the following sections, in particular for the linear elastic case. For brevity, the section on the diffusion problem omits some of the finer details that are completely analogous to the linear elastic case.

2.1 Linear elasticity in the sub-scale

Since the mechanical performance of the SBE will be evaluated, the standard equilibrium problem for a body of isotropic linear elastic material is described. For an RVE with the spatial domain $\Omega_\square$ and boundary $\Gamma_\square$ in the sub-scale, the strong format of the standard quasistatic equilibrium problem for linear elasticity [10] is defined as:

$$-\boldsymbol{\sigma} \cdot \boldsymbol{\nabla} = \boldsymbol{f} \text{ in } \Omega_\square \tag{2.1}$$

where the body forces $\boldsymbol{f} = \boldsymbol{0}$ are neglected in this project. The constitutive equation which corresponds to linear elasticity reads:

$$\boldsymbol{\sigma} = \mathsf{E} : \boldsymbol{\epsilon} \tag{2.2}$$

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}[\boldsymbol{u}] := (\boldsymbol{u} \otimes \boldsymbol{\nabla})^{\text{sym}} \tag{2.3}$$

where E is the constant stiffness tensor defined as:

$$\mathsf{E} = 2G\mathsf{I}^{\text{sym}}_{\text{dev}} + K\boldsymbol{I} \otimes \boldsymbol{I} \tag{2.4}$$

$$\mathsf{I}^{\text{sym}}_{\text{dev}} := \mathsf{I}^{\text{sym}} - \frac{1}{3}\boldsymbol{I} \otimes \boldsymbol{I} \tag{2.5}$$

$$\mathbf{I}^{\text{sym}} := \frac{1}{2}[\boldsymbol{I} \otimes \boldsymbol{I} + \boldsymbol{I} \underline{\otimes} \boldsymbol{I}] \tag{2.6}$$

and the operator $\underline{\otimes}$ is defined as:

$$\boldsymbol{I} \underline{\otimes} \boldsymbol{I} := I_{il} I_{jk} e_i \otimes e_j \otimes e_k \otimes e_l \tag{2.7}$$

The standard Dirichlet and Neumann boundary conditions are:

$$\boldsymbol{u} = \boldsymbol{u}_{\text{P}} \text{ on } \Gamma_{\square,\text{D}} \tag{2.8}$$

$$\boldsymbol{t} := \boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}_{\text{P}} \text{ on } \Gamma_{\square,\text{N}} \tag{2.9}$$

In order to obtain the weak form, the strong form (2.1) is multiplied by a test function $\delta \boldsymbol{u}$ and integrated over the domain $\Omega_\square$. By using the product rule, Gauss's divergence theorem, the Neumann boundary condition (2.9), the symmetric property of $\boldsymbol{\sigma}$ and the constitutive equation (2.2), the following weak form is obtained:

Find $\forall \boldsymbol{u} \in \mathbb{U}$ such that

$$a(\boldsymbol{u}, \delta\boldsymbol{u}) = l(\delta\boldsymbol{u}) \quad \forall \delta\boldsymbol{u} \in \mathbb{U}^0 \tag{2.10}$$

where the introduced functionals are defined as:

$$a(\boldsymbol{u}, \delta\boldsymbol{u}) := \int_{\Omega_\square} \boldsymbol{\epsilon}[\boldsymbol{u}] : \boldsymbol{\mathsf{E}} : \boldsymbol{\epsilon}[\delta\boldsymbol{u}] \; \mathrm{d}\Omega \tag{2.11}$$

$$l(\delta\boldsymbol{u}) := \int_{\Omega_\square} \boldsymbol{f} \cdot \delta\boldsymbol{u} \; \mathrm{d}\Omega + \int_{\Gamma_{\square,\mathrm{N}}} \boldsymbol{t}_\mathrm{P} \cdot \delta\boldsymbol{u} \; \mathrm{d}\Gamma \tag{2.12}$$

and the introduced spaces as:

$$\mathbb{U} = \left\{\boldsymbol{u} : \boldsymbol{u} \in \mathbb{H}^1(\Omega_\square), \boldsymbol{u} = \boldsymbol{u}_\mathrm{P} \text{ on } \Gamma_{\square,\mathrm{D}}\right\} \tag{2.13}$$

$$\mathbb{U}^0 = \left\{\boldsymbol{u} : \boldsymbol{u} \in \mathbb{H}^1(\Omega_\square), \boldsymbol{u} = \boldsymbol{0} \text{ on } \Gamma_{\square,\mathrm{D}}\right\} \tag{2.14}$$

Note that the Sobolev space $\mathbb{H}^1(\Omega) = \{\boldsymbol{u} : \int_\Omega |\boldsymbol{u}|^2 + |\boldsymbol{u} \otimes \boldsymbol{\nabla}|^2 \; \mathrm{d}\Omega < \infty\}$ imposes both differentiability and integrability requirements such that even functions that are only piecewise differentiable can be used.

2.2 Linear elasticity in the macro-scale

Let the linear elastic problem described in Section 2.1 be the sub-scale problem defined by an RVE occupying $\Omega_\square$ and $\Gamma_\square$, then the corresponding macro-scale problem analogously becomes:

$$-\bar{\boldsymbol{\sigma}} \cdot \boldsymbol{\nabla} = \bar{\boldsymbol{f}} \text{ in } \Omega \tag{2.15}$$

$$\bar{\boldsymbol{\sigma}} = \bar{\boldsymbol{\mathsf{E}}} : \bar{\boldsymbol{\epsilon}} \tag{2.16}$$

$$\bar{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}[\bar{\boldsymbol{u}}] := (\bar{\boldsymbol{u}} \otimes \boldsymbol{\nabla})^{\mathrm{sym}} \tag{2.17}$$

where the body forces $\bar{\boldsymbol{f}} = \boldsymbol{0}$ are neglected in this project. The standard Dirichlet and Neumann boundary conditions are:

$$\bar{\boldsymbol{u}} = \bar{\boldsymbol{u}}_\mathrm{P} \text{ on } \Gamma_\mathrm{D} \tag{2.18}$$

$$\bar{\boldsymbol{t}} := \bar{\boldsymbol{\sigma}} \cdot \boldsymbol{n} = \bar{\boldsymbol{t}}_\mathrm{P} \text{ on } \Gamma_\mathrm{N} \tag{2.19}$$

The weak form becomes:

Find $\forall \bar{\boldsymbol{u}} \in \bar{\mathbb{U}}$ such that

$$\bar{a}(\bar{\boldsymbol{u}}, \delta\bar{\boldsymbol{u}}) = \bar{l}(\delta\bar{\boldsymbol{u}}) \quad \forall \delta\bar{\boldsymbol{u}} \in \bar{\mathbb{U}}^0 \tag{2.20}$$

where the introduced functionals are defined as:

$$\bar{a}(\bar{\boldsymbol{u}}, \delta\bar{\boldsymbol{u}}) := \int_{\Omega} \bar{\boldsymbol{\epsilon}}[\bar{\boldsymbol{u}}] : \bar{\boldsymbol{\mathsf{E}}} : \bar{\boldsymbol{\epsilon}}[\delta\bar{\boldsymbol{u}}] \; \mathrm{d}\Omega \tag{2.21}$$

$$\bar{l}(\delta\bar{\boldsymbol{u}}) := \int_{\Omega} \bar{\boldsymbol{f}} \cdot \delta\bar{\boldsymbol{u}} \; \mathrm{d}\Omega + \int_{\Gamma_\mathrm{N}} \bar{\boldsymbol{t}}_\mathrm{P} \cdot \delta\bar{\boldsymbol{u}} \; \mathrm{d}\Gamma \tag{2.22}$$

and the introduced spaces as:

$$\bar{\mathbb{U}} = \left\{\boldsymbol{u} : \boldsymbol{u} \in \mathbb{H}^1(\Omega), \boldsymbol{u} = \bar{\boldsymbol{u}}_\mathrm{P} \text{ on } \Gamma_\mathrm{D}\right\} \tag{2.23}$$

$$\bar{\mathbb{U}}^0 = \left\{\boldsymbol{u} : \boldsymbol{u} \in \mathbb{H}^1(\Omega), \boldsymbol{u} = \boldsymbol{0} \text{ on } \Gamma_\mathrm{D}\right\} \tag{2.24}$$

Note that the functionals and spaces are analogous to the sub-scale case.

2.3 Computation of effective stiffness

Now that the upscaling of linear elasticity is formulated, the next step is to apply the prolongation rule which is a scale separation method which employs an additive split of the displacement field:

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}^{\mathrm{M}}(\boldsymbol{x}) + \boldsymbol{u}^{\mathrm{S}}(\boldsymbol{x}) \quad \boldsymbol{x} \in \Omega_{\square} \tag{2.25}$$

Here, $\boldsymbol{u}(\boldsymbol{x})$ is the sub-scale displacement field, $\boldsymbol{u}^{\mathrm{M}}(\boldsymbol{x})$ is the smooth displacement field, and $\boldsymbol{u}^{\mathrm{S}}(\boldsymbol{x})$ is the sub-scale displacement fluctuation field. In the same way, the strain is decomposed as:

$$\boldsymbol{\epsilon}(\boldsymbol{x}) = \bar{\boldsymbol{\epsilon}} + \boldsymbol{\epsilon}^{\mathrm{S}}(\boldsymbol{x}) \tag{2.26}$$

where:

$$\boldsymbol{\epsilon}^{\mathrm{S}} = \boldsymbol{\epsilon}\left[\boldsymbol{u}^{\mathrm{S}}\right] \tag{2.27}$$

The smooth displacement field is known since it is prescribed according to the assumption of first order homogenization:

$$\boldsymbol{u}^{\mathrm{M}}(\boldsymbol{x}) = \bar{\boldsymbol{u}} + \bar{\boldsymbol{\epsilon}} \cdot [\boldsymbol{x} - \bar{\boldsymbol{x}}] \qquad \boldsymbol{x} \in \Omega_{\square} \tag{2.28}$$

where the user-defined macro-scale strain $\bar{\boldsymbol{\epsilon}}$ serves as the driving force for the problem and the macro-scale displacement field $\bar{\boldsymbol{u}}$ can simply be put to zero in order to remove rigid body motion. The reference point $\bar{\boldsymbol{x}}$ is chosen arbitrarily, but a common choice is the RVE centre. Together with the necessary material parameters and boundary conditions, the weak form can be solved, whereby the sub-scale stress is obtained. Volume averaging the sub-scale stress as a post-processing step gives the macro-scale stress:

$$\bar{\boldsymbol{\sigma}} = \langle \boldsymbol{\sigma} \rangle_{\square} \tag{2.29}$$

where the volume average operator is defined as:

$$\langle \bullet \rangle_{\square} := \frac{1}{|\Omega_{\square}|} \int_{\Omega_{\square}} \bullet \, \mathrm{d}\Omega \tag{2.30}$$

With known macro-scale stress and strain, it becomes easy to compute the effective macro-scale stiffness, see Figure 2.1 for an overview of the whole process.

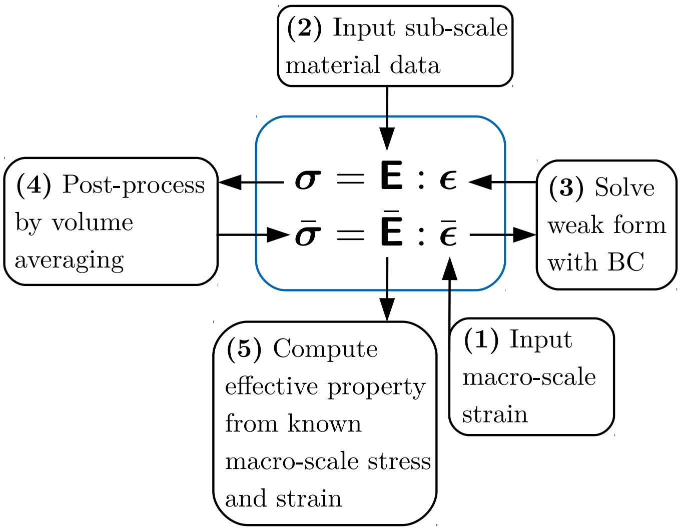

(2) Input sub-scale material data
$\boldsymbol{\sigma} = \mathbf{E} : \boldsymbol{\epsilon}$
$\bar{\boldsymbol{\sigma}} = \bar{\mathbf{E}} : \bar{\boldsymbol{\epsilon}}$
(4) Post-process by volume averaging
(3) Solve weak form with BC
(5) Compute effective property from known macro-scale stress and strain
(1) Input macro-scale strain

Figure 2.1: Overview of the computation of effective stiffness.

One crucial constraint that has not yet been discussed is the Hill-Mandel macrohomogeneity condition which serves as a link between the macro-scale variables $\{\bar{\boldsymbol{\sigma}}, \bar{\boldsymbol{\epsilon}}\}$ and the sub-scale variables $\{\boldsymbol{\sigma}, \boldsymbol{\epsilon}\}$. For the case of macro-scale strain control, two types of constraints must hold a priori:

$$\langle \boldsymbol{\epsilon} \rangle_{\Box} = \bar{\boldsymbol{\epsilon}} \tag{2.31}$$

which is the strain identity, and also:

$$\langle \boldsymbol{\sigma} : \boldsymbol{\epsilon} \rangle_{\Box} = \langle \boldsymbol{\sigma} \rangle_{\Box} : \langle \boldsymbol{\epsilon} \rangle_{\Box} \tag{2.32}$$

which is the work identity. After rewriting the Hill-Mandel macrohomogeneity condition, it can be summarized in a neat fashion:

$$\int_{\Gamma_{\Box}} [\boldsymbol{t} - \bar{\boldsymbol{\sigma}} \cdot \boldsymbol{n}] \cdot [\boldsymbol{u} - \bar{\boldsymbol{\epsilon}} \cdot [\boldsymbol{x} - \bar{\boldsymbol{x}}]] \, \mathrm{d}\Gamma = 0 \tag{2.33}$$

where $\boldsymbol{t}^{\mathrm{M}} := \bar{\boldsymbol{\sigma}} \cdot \boldsymbol{n}$ and $\boldsymbol{u}^{\mathrm{S}} = [\boldsymbol{u} - \bar{\boldsymbol{\epsilon}} \cdot [\boldsymbol{x} - \bar{\boldsymbol{x}}]]$. See the compendium [12] by R. Jänicke, F. Larsson and K. Runesson for more information.

2.4 Weakly periodic boundary conditions for linear elasticity

So far, the sub-scale problem defined by the RVE is still not solvable since the boundary conditions are not fully specified. There are several standard choices for prescribing the boundary conditions on $\Gamma_{\Box}$ in order to estimate $\bar{\mathsf{E}}$, but the challenge is to choose the boundary conditions which give the most accurate estimation. The standard choices are Dirichlet boundary conditions (DBC), Neumann boundary conditions (NBC) and Periodic boundary conditions (PBC). See the compendium [12] by R. Jänicke, F. Larsson and K. Runesson for more information.

In this project, the weakly periodic boundary conditions (WPBC) are used for the finite element simulations, but first, the strongly periodic boundary conditions (SPBC) need to be introduced. One of the most crucial aspects of the SPBC is fulfilling the micro-periodicity assumption in a strong sense. Micro-periodicity assumes that the sub-scale fluctuation field $\boldsymbol{u}^{\mathrm{S}}$ is periodic on the boundary of the RVE. An RVE can in general have any arbitrary shape, but for simplicity and convenience, they often take the shape of a 2D unit square or a 3D unit cube. For a 2D square RVE, micro-periodicity results in that the left and the right RVE boundaries must have the same fluctuation field, and the same goes for the bottom and the top RVE boundaries. While a 2D square RVE has 2 pairs of periodic boundaries, a 3D cube RVE has 3 pairs of periodic boundaries. The micro-periodicity assumption comes from the perception that all materials consist of repetitive RVEs with the same shape and content [13]. This implies that even if several RVEs are pieced together, they still remain kinematically compatible during deformation. Before employing the micro-periodicity assumption, it is first necessary to introduce the following boundary split:

$$\Gamma_{\Box} = \Gamma_{\Box}^{+} \cup \Gamma_{\Box}^{-} \tag{2.34}$$

where $\Gamma_{\Box}^{+}$ is the image boundary (positive side) and $\Gamma_{\Box}^{-}$ is the mirror boundary (negative side), see Figure 2.2.

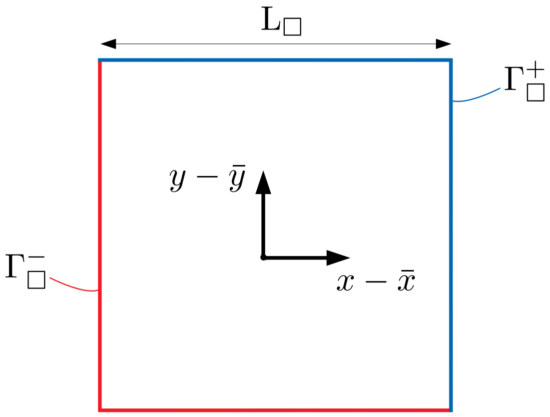


Figure 2.2: Splitting the boundary of a 2D square RVE into an image and mirror boundary.

The consequence of introducing such a boundary split is that each and every point on the boundary gets a partner point on the opposite side. However, this does not apply for the corner points since they have 2 partner points each for a 2D square RVE, and 3 partner points each for a 3D cube RVE. Due to this, the corner points are not included in the image and mirror boundaries. The solution is that they form their own definition on $\Gamma_\square$ and are dealt with separately:

$$\boldsymbol{x}_\mathrm{c}^{i,j,k} = L_\square \cdot [i\boldsymbol{e}_1 + j\boldsymbol{e}_2 + k\boldsymbol{e}_3] \tag{2.35}$$

where $L_\square$ is the RVE side length and $i, j, k = 1$ or 0 for a 3D cube RVE with the origin in the corner. The next step is to introduce the periodic mapping operator:

$$\varphi_\mathrm{per} : \Gamma_\square^+ \mapsto \Gamma_\square^- \tag{2.36}$$

such that:

$$\boldsymbol{x}^- = \varphi_\mathrm{per}(\boldsymbol{x}^+) \tag{2.37}$$

Finally, the micro-periodicity of the displacement fluctuation field can be expressed as:

$$\boldsymbol{u}^\mathrm{S}(\boldsymbol{x}) = \boldsymbol{u}^\mathrm{S}(\varphi_\mathrm{per}(\boldsymbol{x})) \quad \forall \boldsymbol{x} \in \Gamma_\square^+ \tag{2.38}$$

A neat way to express this is as follows:

$$[\![\boldsymbol{u}^\mathrm{S}]\!]_\square = \boldsymbol{0} \quad \forall \boldsymbol{x} \in \Gamma_\square^+ \tag{2.39}$$

where the jump operator between the image and mirror boundaries is defined as:

$$[\![\boldsymbol{u}^\mathrm{S}]\!]_\square(\boldsymbol{x}) \coloneqq \boldsymbol{u}^\mathrm{S}(\boldsymbol{x}) - \boldsymbol{u}^\mathrm{S}(\varphi_\mathrm{per}(\boldsymbol{x})) \quad \forall \boldsymbol{x} \in \Gamma_\square^+ \tag{2.40}$$

Due to the introduction of the jump operator, the image boundary is henceforth chosen as the main computation domain for boundary integration. Another condition that needs to be fulfilled is the symmetry condition of the sub-scale stress:

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{\sigma}(\varphi_\mathrm{per}(\boldsymbol{x})) \quad \forall \boldsymbol{x} \in \Gamma_\square^+ \tag{2.41}$$

which in turn automatically leads to the anti-periodicity condition for the traction:

$$\boldsymbol{t}(\boldsymbol{x}) = -\boldsymbol{t}(\varphi_\mathrm{per}(\boldsymbol{x})) \quad \forall \boldsymbol{x} \in \Gamma_\square^+ \tag{2.42}$$

Finally, the last SPBC-constraint to fulfill is the periodicity condition for the corners which results in that all corner points get the same displacement fluctuation field:

$$\boldsymbol{u}^\mathrm{S}(\boldsymbol{x}_\mathrm{c}^{i,j,k}) = \boldsymbol{u}_\mathrm{c}^\mathrm{S} \tag{2.43}$$

With all of this, the SPBC are properly defined and the micro-periodicity can be prescribed in a strong sense. A crucial consequence of SPBC is that the Hill-Mandel macrohomogeneity condition (2.33) becomes trivially satisfied:

$$\begin{aligned}
\int_{\Gamma_\square} [\boldsymbol{t}(\boldsymbol{x}) - \bar{\boldsymbol{\sigma}}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x})] \cdot \boldsymbol{u}^\mathrm{S}(\boldsymbol{x})\ \mathrm{d}\Gamma &= \int_{\Gamma_\square^+} [\boldsymbol{t}(\boldsymbol{x}) - \boldsymbol{t}^\mathrm{M}(\boldsymbol{x})] \cdot \boldsymbol{u}^\mathrm{S}(\boldsymbol{x})\ \mathrm{d}\Gamma + \int_{\Gamma_\square^-} [\boldsymbol{t}(\boldsymbol{x}) - \boldsymbol{t}^\mathrm{M}(\boldsymbol{x})] \cdot \boldsymbol{u}^\mathrm{S}(\boldsymbol{x})\ \mathrm{d}\Gamma \\
&= \int_{\Gamma_\square^+} [\boldsymbol{t}(\boldsymbol{x}) - \boldsymbol{t}^\mathrm{M}(\boldsymbol{x})] \cdot \boldsymbol{u}^\mathrm{S}(\boldsymbol{x})\ \mathrm{d}\Gamma + \int_{\Gamma_\square^+} [\boldsymbol{t}(\varphi_\mathrm{per}(\boldsymbol{x})) - \boldsymbol{t}^\mathrm{M}(\varphi_\mathrm{per}(\boldsymbol{x}))] \cdot \boldsymbol{u}^\mathrm{S}(\varphi_\mathrm{per}(\boldsymbol{x}))\ \mathrm{d}\Gamma \\
&= \int_{\Gamma_\square^+} [\boldsymbol{t}(\boldsymbol{x}) - \boldsymbol{t}^\mathrm{M}(\boldsymbol{x})] \cdot \boldsymbol{u}^\mathrm{S}(\boldsymbol{x}) + [-\boldsymbol{t}(\boldsymbol{x}) + \boldsymbol{t}^\mathrm{M}(\boldsymbol{x})] \cdot \boldsymbol{u}^\mathrm{S}(\varphi_\mathrm{per}(\boldsymbol{x}))\ \mathrm{d}\Gamma \\
&= \int_{\Gamma_\square^+} [\boldsymbol{t} - \boldsymbol{t}^\mathrm{M}] \cdot [\![\boldsymbol{u}^\mathrm{S}]\!]\ \mathrm{d}\Gamma \\
&= 0
\end{aligned} \tag{2.44}$$

where the micro-periodicity condition $[\![\boldsymbol{u}^\mathrm{S}]\!] = 0$ from Equation (2.39) is used.

However, a way to relax the strict constraints of SPBC is to introduce the WPBC which impose the micro-periodicity in a weak sense. Consider the general sub-scale weak form (2.10) in a slightly rewritten format:

$$\frac{1}{|\Omega_\square|}\int_{\Omega_\square} \boldsymbol{\epsilon}[\boldsymbol{u}] : \mathsf{E} : \boldsymbol{\epsilon}[\delta\boldsymbol{u}] \,\mathrm{d}\Omega - \frac{1}{|\Omega_\square|}\int_{\Gamma_{\square,\mathrm{N}}} \boldsymbol{t}_\mathrm{P}\cdot\delta\boldsymbol{u}\,\mathrm{d}\Gamma = 0 \tag{2.45}$$

Using the anti-periodicity of the traction (2.42) on the boundary term from (2.45) yields:

$$\begin{aligned}
\frac{1}{|\Omega_\square|}\int_{\Gamma_{\square,\mathrm{N}}} \boldsymbol{t}_\mathrm{P}(\boldsymbol{x})\cdot\delta\boldsymbol{u}(\boldsymbol{x})\,\mathrm{d}\Gamma &= \frac{1}{|\Omega_\square|}\left[\int_{\Gamma^+_{\square,\mathrm{N}}} \boldsymbol{t}_\mathrm{P}(\boldsymbol{x})\cdot\delta\boldsymbol{u}(\boldsymbol{x})\,\mathrm{d}\Gamma + \int_{\Gamma^-_{\square,\mathrm{N}}} \boldsymbol{t}_\mathrm{P}(\boldsymbol{x})\cdot\delta\boldsymbol{u}(\boldsymbol{x})\,\mathrm{d}\Gamma\right] \\
&= \frac{1}{|\Omega_\square|}\left[\int_{\Gamma^+_{\square,\mathrm{N}}} \boldsymbol{t}_\mathrm{P}(\boldsymbol{x})\cdot\delta\boldsymbol{u}(\boldsymbol{x})\,\mathrm{d}\Gamma + \int_{\Gamma^+_{\square,\mathrm{N}}} \boldsymbol{t}_\mathrm{P}(\varphi_\mathrm{per}(\boldsymbol{x}))\cdot\delta\boldsymbol{u}(\varphi_\mathrm{per}(\boldsymbol{x}))\,\mathrm{d}\Gamma\right] \\
&= \frac{1}{|\Omega_\square|}\left[\int_{\Gamma^+_{\square,\mathrm{N}}} \boldsymbol{t}_\mathrm{P}(\boldsymbol{x})\cdot\delta\boldsymbol{u}(\boldsymbol{x}) - \boldsymbol{t}_\mathrm{P}(\boldsymbol{x})\cdot\delta\boldsymbol{u}(\varphi_\mathrm{per}(\boldsymbol{x}))\,\mathrm{d}\Gamma\right] \\
&= \frac{1}{|\Omega_\square|}\left[\int_{\Gamma^+_{\square,\mathrm{N}}} \boldsymbol{t}_\mathrm{P}\cdot[\![\delta\boldsymbol{u}]\!]\,\mathrm{d}\Gamma\right]
\end{aligned} \tag{2.46}$$

Furthermore, the strong micro-periodicity constraint from (2.39) is formulated in a weak sense for WPBC:

$$\int_{\Gamma^+_{\square,\mathrm{N}}} \delta\boldsymbol{t}\cdot[\![\boldsymbol{u}^\mathrm{S}]\!]\,\mathrm{d}\Gamma = 0 \quad \forall\delta\boldsymbol{t}\in\mathbb{T}^+_\square \tag{2.47}$$

where the traction $\boldsymbol{t}$ corresponds to the Lagrangian multiplier field $\boldsymbol{\lambda}$. Finally, the weak form of the WPBC problem for linear elasticity can be formulated as:

Find $\forall\boldsymbol{u}\in\mathbb{U}_\square$ and $\forall\boldsymbol{t}\in\mathbb{T}^+_\square$ for given macro-scale strain $\bar{\boldsymbol{\epsilon}}$ such that

$$a_\square(\boldsymbol{u};\delta\boldsymbol{u}) - d_\square(\boldsymbol{t},\delta\boldsymbol{u}) = 0 \qquad \forall\delta\boldsymbol{u}\in\mathbb{U}_\square \tag{2.48}$$

$$-d_\square(\delta\boldsymbol{t},\boldsymbol{u}) = -d_\square(\delta\boldsymbol{t},\bar{\boldsymbol{\epsilon}}\cdot[\boldsymbol{x}-\bar{\boldsymbol{x}}]) \quad \forall\delta\boldsymbol{t}\in\mathbb{T}^+_\square \tag{2.49}$$

where the introduced functionals are defined as:

$$a_\square(\boldsymbol{u},\delta\boldsymbol{u}) \coloneqq \frac{1}{|\Omega_\square|}\int_{\Omega_\square} \boldsymbol{\epsilon}[\boldsymbol{u}] : \mathsf{E} : \boldsymbol{\epsilon}[\delta\boldsymbol{u}]\,\mathrm{d}\Omega \tag{2.50}$$

$$d_\square(\boldsymbol{t},\boldsymbol{u}) \coloneqq \frac{1}{|\Omega_\square|}\int_{\Gamma^+_\square} \boldsymbol{t}\cdot[\![\boldsymbol{u}]\!]_\square\,\mathrm{d}\Gamma \tag{2.51}$$

and $\mathbb{T}^+_\square$ is the trace of functions in the self-equilibrating traction space $\mathbb{T}_\square$ on $\Gamma^+_\square$:

$$\mathbb{T}_\square = \left\{\boldsymbol{t} : \boldsymbol{t} \text{ sufficiently regular in } \Omega_\square, \int_{\Gamma_\square} \boldsymbol{t}\,\mathrm{d}\Gamma = \boldsymbol{0}\right\} \tag{2.52}$$

$$\mathbb{U}_\square = \left\{\boldsymbol{u} : \boldsymbol{u}\in\mathbb{H}^1(\Omega_\square), \int_{\Omega_\square} \boldsymbol{u}\,\mathrm{d}\Omega = \boldsymbol{0}, \int_{\Gamma_\square} (\boldsymbol{u}\otimes\boldsymbol{n})^\mathrm{skw}\,\mathrm{d}\Gamma = \boldsymbol{0}\right\} \tag{2.53}$$

2.5 Fluid–structure interaction in SBE

For the linear elastic problem, it is unclear how to determine the internal pressure when the polymer matrix deforms and compresses the liquid electrolyte in the pore space. Since solving the fully coupled structural and CFD problem is beyond the scope of this project, a more feasible approach is to only consider the polymer matrix and to account for the fluid–structure interaction by adding additional global constraint equations to the weak form.

Consider two extreme scenarios that serve as lower and upper bounds. The so called drained case corresponds to when the liquid electrolyte is free to escape from the pores without any resistance, i.e. no internal fluid pressure acts on the polymer matrix during loading. This scenario gives the lower bound of the effective stiffness. The upper bound is given by the so called undrained case where the liquid electrolyte is assumed to be completely trapped inside the pore space by a very stiff membrane, thus an internal fluid pressure acts on the polymer matrix during loading since the liquid electrolyte is assumed to be incompressible. The reason this scenario serves as the upper bound is because the internal fluid pressure contributes to the effective stiffness. The reality however, must be a case that lies somewhere in between these extreme cases, i.e. the liquid electrolyte must be trapped by a membrane that has a certain reasonable stiffness.

Since the drained case is not affected by the fluid, no modifications to the current weak form (2.48,2.49) are necessary. However, the undrained case requires special treatment since there is an internal fluid pressure that acts on the interface between the solid polymer matrix and liquid electrolyte, see Figure 2.3.

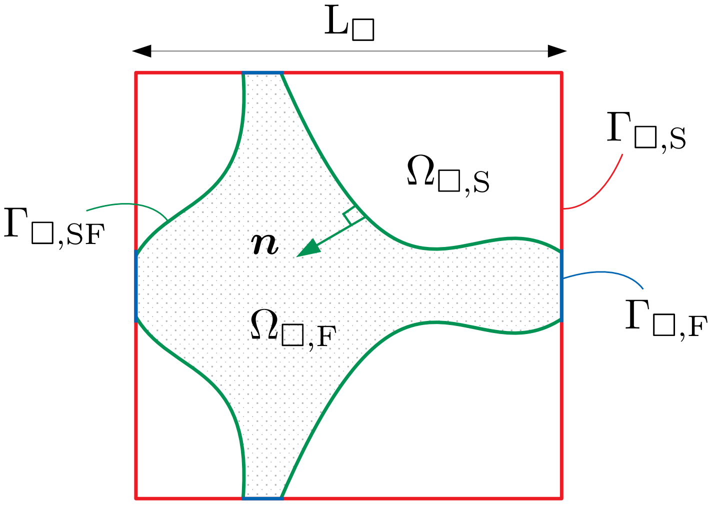


Figure 2.3: Special domain and boundary split of a periodic 2D square RVE for undrained elasticity. Subscripts: S = Solid, F = Fluid and SF = Solid-Fluid interface.

For the undrained case, both the solid and fluid phase have to satisfy the equilibrium equation (2.1):

$$-\boldsymbol{\sigma}\cdot\boldsymbol{\nabla}=\mathbf{0} \text{ in } \Omega_\square=\Omega_{\square,\mathrm{S}}\cup\Omega_{\square,\mathrm{F}} \tag{2.54}$$

$$\boldsymbol{\sigma}=\mathbf{E}:\boldsymbol{\epsilon} \text{ in } \Omega_{\square,\mathrm{S}} \tag{2.55}$$

$$\boldsymbol{\sigma}=-\bar{\lambda}\cdot\boldsymbol{I} \text{ in } \Omega_{\square,\mathrm{F}} \tag{2.56}$$

where the added unknown degree of freedom (DOF) $\bar{\lambda}$ is the internal pressure. Using the current weak form (2.48,2.49) as a starting point and inserting the new split of the domains and stresses (2.55,2.56) gives the weak form for undrained elasticity:

Find $\forall \boldsymbol{u} \in \mathbb{U}_{\square}$, $\forall \boldsymbol{t} \in \mathbb{T}_{\square}^{+}$ and $\forall \bar{\lambda} \in \mathbb{R}$ for given macro-scale strain $\bar{\boldsymbol{\epsilon}}$ such that:

$$\frac{1}{|\Omega_{\square}|}\int_{\Omega_{\square,\mathrm{S}}} \boldsymbol{\epsilon}[\boldsymbol{u}] : \mathbf{E} : \boldsymbol{\epsilon}[\delta\boldsymbol{u}]\ \mathrm{d}\Omega + \frac{1}{|\Omega_{\square}|}\int_{\Omega_{\square,\mathrm{S}}} \bar{\lambda}\boldsymbol{I} : \boldsymbol{\epsilon}[\delta\boldsymbol{u}]\ \mathrm{d}\Omega - \frac{1}{|\Omega_{\square}|}\int_{\Gamma_{\square,\mathrm{S}}^{+}} \boldsymbol{t}\cdot [\![\delta\boldsymbol{u}]\!]_{\square}\ \mathrm{d}\Gamma = 0 \qquad \forall \delta\boldsymbol{u} \in \mathbb{U}_{\square} \tag{2.57}$$

$$-\frac{1}{|\Omega_{\square}|}\int_{\Gamma_{\square,\mathrm{S}}^{+}} \delta\boldsymbol{t}\cdot [\![\boldsymbol{u}]\!]_{\square}\ \mathrm{d}\Gamma = -\frac{1}{|\Omega_{\square}|}\int_{\Gamma_{\square,\mathrm{S}}^{+}} \delta\boldsymbol{t}\otimes [\![\boldsymbol{x}]\!]_{\square}\ \mathrm{d}\Gamma : \bar{\boldsymbol{\epsilon}} \qquad \forall \delta\boldsymbol{t} \in \mathbb{T}_{\square}^{+} \tag{2.58}$$

$$\delta\bar{\lambda}\left[\frac{1}{|\Omega_{\square}|}\int_{\Gamma_{\square,\mathrm{SF}}} \boldsymbol{u}\cdot\boldsymbol{n}\ \mathrm{d}\Gamma + \frac{1}{|\Omega_{\square}|}\int_{\Gamma_{\square,\mathrm{S}}^{+}} [\![\boldsymbol{u}]\!]_{\square}\cdot\boldsymbol{n}\ \mathrm{d}\Gamma\right] = \delta\bar{\lambda}\boldsymbol{I} : \bar{\boldsymbol{\epsilon}} \qquad \forall \delta\bar{\lambda} \in \mathbb{R} \tag{2.59}$$

where $\mathbb{R}$ corresponds to the set of real numbers.

Note that the undrained case results in an extra contribution to the weak form and one additional global constraint equation. These changes correspond to imposing an incompressibility constraint on the fluid.

2.6 Stationary diffusion and heat flow in the sub-scale

The standard diffusion problem is formulated in order to evaluate the ionic conductivity of the SBE. Note that only the fluid phase in the pore space is considered for the diffusion problem since the ions are transported via the liquid electrolyte. Although the thermal effects are not taken into account in this thesis, the heat equation plays an important role in the generation of the artificial SBE microstructures. The details of how the heat equation is utilized are described in Section 4.2. The governing equations for diffusion and heat flow have the same structure, hence it is convenient to describe the equations using generic variables.

Consider the strong form of a generic stationary (steady-state) continuity equation [11] which balances the flux- and the source contributions per unit volume and per unit time in a sub-scale domain:

$$\boldsymbol{\nabla}\cdot\boldsymbol{q} = Q \text{ in } \Omega_{\square} \tag{2.60}$$

where the internal source $Q = 0$ for the diffusion problem. The corresponding generic constitutive relation is defined as:

$$\boldsymbol{q} = -\boldsymbol{D}\cdot\boldsymbol{\nabla}u \tag{2.61}$$

where Fick's first law is used for chemical diffusion and Fourier's law of thermal conduction is used for heat flow. The generic variables will have different definitions depending on the type of problem that is considered, see Table 2.1.

Table 2.1: Definition of generic variables for chemical diffusion and heat flow.

	Chemical diffusion	**Heat flow**
$\boldsymbol{q}$	Ion flux	Heat flux
Q	Internal ion supply	Internal heat supply
$\boldsymbol{D}$	Diffusion coefficient matrix	Thermal conductivity matrix
u	Ion concentration	Temperature

The standard Dirichlet and Neumann boundary conditions are:

$$u = g \text{ on } \Gamma_{\square,\mathrm{D}} \tag{2.62}$$

$$q_{\mathrm{n}} := \boldsymbol{q}\cdot\boldsymbol{n} = h \text{ on } \Gamma_{\square,\mathrm{N}} \tag{2.63}$$

In a similar way as for linear elasticity, the weak form is obtained by multiplying the strong form (2.60) by a test function δu and integrating over the domain $\Omega_{\square}$. Using the product rule, Gauss's divergence theorem, the

Neumann boundary condition (2.63), and the constitutive equation (2.61), the following weak form is obtained:

Find $\forall u \in \mathbb{V}$ that solves

$$\hat{a}(u, \delta u) = \hat{l}(\delta u) \quad \forall \delta u \in \mathbb{V}^0 \tag{2.64}$$

where the introduced functionals are defined as:

$$\hat{a}(u, \delta u) := \int_{\Omega_\square} [\boldsymbol{\nabla} u] \cdot \boldsymbol{D} \cdot [\boldsymbol{\nabla} \delta u] \; \mathrm{d}\Omega \tag{2.65}$$

$$\hat{l}(\delta u) := \int_{\Omega_\square} Q \cdot \delta u \; \mathrm{d}\Omega - \int_{\Gamma_{\square,\mathrm{N}}} q_{\mathrm{n}} \cdot \delta u \; \mathrm{d}\Gamma \tag{2.66}$$

and the introduced spaces as:

$$\mathbb{V} = \left\{ u : u \in \mathbb{H}^1(\Omega_\square), u = u_{\mathrm{P}} \text{ on } \Gamma_{\square,\mathrm{D}} \right\} \tag{2.67}$$

$$\mathbb{V}^0 = \left\{ u : u \in \mathbb{H}^1(\Omega_\square), u = 0 \text{ on } \Gamma_{\square,\mathrm{D}} \right\} \tag{2.68}$$

2.7 Stationary diffusion in the macro-scale

Once again, let the diffusion problem described in Section 2.6 be the sub-scale problem defined by the RVE in $\Omega_\square$ and $\Gamma_\square$, then the corresponding upscaled macro-scale problem analogously becomes:

$$\boldsymbol{\nabla} \cdot \bar{\boldsymbol{q}} = \bar{Q} \text{ in } \Omega \tag{2.69}$$

$$\bar{\boldsymbol{q}} = -\bar{\boldsymbol{D}} \cdot \boldsymbol{\nabla} \bar{u} \tag{2.70}$$

where the internal source $\bar{Q} = 0$ for the diffusion problem. The standard Dirichlet and Neumann boundary conditions are:

$$\bar{u} = \bar{g} \text{ on } \Gamma_{\mathrm{D}} \tag{2.71}$$

$$\bar{q}_n := \bar{\boldsymbol{q}} \cdot \boldsymbol{n} = \bar{h} \text{ on } \Gamma_{\mathrm{N}} \tag{2.72}$$

The weak form becomes:

Find $\forall \bar{u} \in \mathbb{U}$ such that

$$\hat{\bar{a}}(\bar{u}, \delta \bar{u}) = \hat{\bar{l}}(\delta \bar{u}) \quad \forall \delta \bar{u} \in \bar{\mathbb{U}}^0 \tag{2.73}$$

where the introduced functionals are defined as:

$$\hat{\bar{a}}(\bar{u}, \delta \bar{u}) := \int_{\Omega_\square} [\boldsymbol{\nabla} \bar{u}] \cdot \bar{\boldsymbol{D}} \cdot [\boldsymbol{\nabla} \delta \bar{u}] \; \mathrm{d}\Omega \tag{2.74}$$

$$\hat{\bar{l}}(\delta \bar{u}) := \int_{\Omega} \bar{Q} \cdot \delta \bar{u} \; \mathrm{d}\Omega - \int_{\Gamma_{\mathrm{N}}} \bar{q}_n \cdot \delta \bar{u} \; \mathrm{d}\Gamma \tag{2.75}$$

and the introduced spaces as:

$$\bar{\mathbb{V}} = \left\{ u : u \in \mathbb{H}^1(\Omega), u = \bar{u}_{\mathrm{P}} \text{ on } \Gamma_{\mathrm{D}} \right\} \tag{2.76}$$

$$\bar{\mathbb{V}}^0 = \left\{ u : u \in \mathbb{H}^1(\Omega), u = 0 \text{ on } \Gamma_{\mathrm{D}} \right\} \tag{2.77}$$

Note that the functionals and spaces are analogous to the sub-scale case.

2.8 Computation of effective diffusivity

In the same way as described for linear elasticity in Section 2.3, the prolongation rule is used. The assumption of first order homogenization gives:

$$u^{\mathrm{M}}(\boldsymbol{x}) = \boldsymbol{\nabla} \bar{u} \cdot [\boldsymbol{x} - \bar{\boldsymbol{x}}] \quad \boldsymbol{x} \in \Omega_\square \tag{2.78}$$

where it is assumed that the diffusion problem is invariant for the overall level $\bar{u}$. For convenience, $\bar{u} = 0$ is chosen henceforth. Note that in this case, the macro-scale ion gradient $\boldsymbol{\nabla}\bar{u}$ is the driving force. Together with the necessary material parameters and boundary conditions, the weak form can be solved, whereby the sub-scale ion flux is obtained. Volume averaging the sub-scale ion flux as a post-processing step gives the macro-scale ion flux:

$$\bar{\boldsymbol{q}} = \langle \boldsymbol{q} \rangle_\square \tag{2.79}$$

With known macro-scale ion flux and ion gradient, it becomes easy to compute the effective macro-scale diffusion coefficient matrix, see Figure 2.4 for an overview of the whole process.

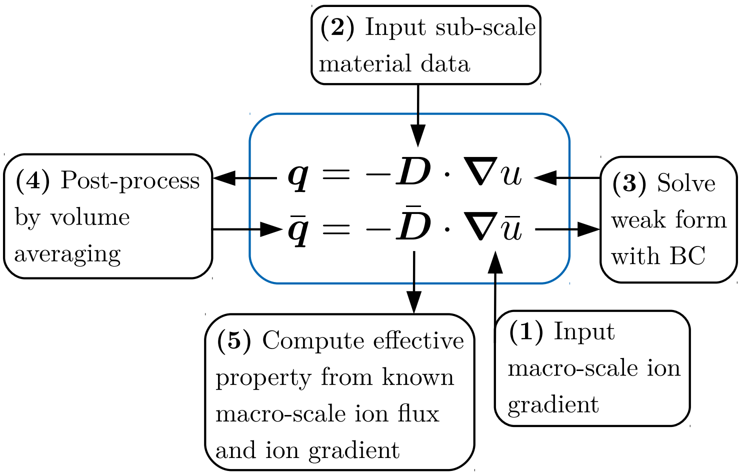

(2) Input sub-scale material data
$\boldsymbol{q} = -\boldsymbol{D} \cdot \boldsymbol{\nabla} u$
$\bar{\boldsymbol{q}} = -\bar{\boldsymbol{D}} \cdot \boldsymbol{\nabla} \bar{u}$
(4) Post-process by volume averaging
(3) Solve weak form with BC
(5) Compute effective property from known macro-scale ion flux and ion gradient
(1) Input macro-scale ion gradient

Figure 2.4: Overview of the computation of effective diffusion coefficient matrix.

2.9 Weakly periodic boundary conditions for stationary diffusion

The details of WPBC for diffusion are omitted, the reader is referred to Section 2.4. The weak form of the WPBC problem for the diffusion equation analogously becomes:

Find $\forall u \in \hat{\mathbb{U}}_\square$ and $\forall q_\mathrm{n} \in \hat{\mathbb{T}}^+_\square$ for given macro-scale ion gradient $\boldsymbol{\nabla}\bar{u}$ such that

$$\hat{a}_\square(u; \delta u) - \hat{d}_\square(q_\mathrm{n}, \delta u) = 0 \qquad \forall \delta u \in \hat{\mathbb{U}}_\square \tag{2.80}$$

$$-\hat{d}_\square(\delta q_\mathrm{n}, u) = -\hat{d}_\square(\delta q_\mathrm{n}, \boldsymbol{\nabla}\bar{u} \cdot [\boldsymbol{x} - \bar{\boldsymbol{x}}]) \quad \forall \delta q_\mathrm{n} \in \hat{\mathbb{T}}^+_\square \tag{2.81}$$

where the introduced functional is defined as:

$$\hat{a}_\square(u, \delta u) := \frac{1}{|\Omega_\square|} \int_{\Omega_\square} [\boldsymbol{\nabla} u] \cdot \boldsymbol{D} \cdot [\boldsymbol{\nabla} \delta u] \, \mathrm{d}\Omega \tag{2.82}$$

$$\hat{d}_\square(q_\mathrm{n}, u) := \frac{1}{|\Omega_\square|} \int_{\Gamma^+_\square} q_\mathrm{n} \cdot [\![u]\!]_\square \, \mathrm{d}\Gamma \tag{2.83}$$

and $\hat{\mathbb{T}}^+_\square$ is the trace of functions in the self-equilibrating normal flux space on $\Gamma^+_\square$:

$$\hat{\mathbb{T}}_\square = \left\{ q_\mathrm{n} : q_\mathrm{n} \text{ sufficiently regular in } \Omega_\square, \int_{\Gamma_\square} q_\mathrm{n} \, \mathrm{d}\Gamma = 0 \right\} \tag{2.84}$$

$$\hat{\mathbb{U}}_\square = \left\{ u : u \in \mathbb{H}^1(\Omega_\square), \int_{\Omega_\square} u \, \mathrm{d}\Omega = 0 \right\} \tag{2.85}$$

3 Voronoi tessellation

The microstructure of real materials can be obtained from 3D measurements such as X-ray computed tomography (CT), and by combining the data from Focused ion beam (FIB) with Scanning electron microscopy (SEM) based on the FIB-SEM method. However, the amount of data in these scans is way too large for the RVE. It is too expensive to perform any FE-analysis on models based on them. An efficient alternative is to generate artificial microstructures and to use them as the RVEs instead. Ideally, the 3D measurement data should be used to identify stochastic structural parameters (e.g. volume fraction and pore size distribution) that can be imposed on the artificial microstructure. This results in an artificial microstructure that is simplified, yet stochastically similar to the real one.

The usage of Voronoi tessellation for generation of artificial microstructure is a well-established method. E.g. the Voronoi tessellation is utilized in the modeling of crystalline aggregates [14], asphalt concrete [15], and even skeletal muscle tissues [16]. In this project, the Voronoi tessellation will form the basis for a foam-like microstructure that fulfills the SBE microstructure constraints, i.e. micro-porosity and bicontinuity.

3.1 Standard Voronoi tessellation

Although it is the 3D Voronoi tessellation that is used for the generation of artificial microstructures, the 2D Voronoi tessellation is more illustrative and easier to explain. Therefore, the examples given here will pertain to the 2D case since extending the theory to 3D is straightforward. The 2D Voronoi tessellation partitions the 2D domain into convex polygons called Voronoi cells based on n user-defined seeds $\{\boldsymbol{p}_1, ..., \boldsymbol{p}_n\}$. In essence, each Voronoi cell is a set of points that is defined as follows:

$$V_i = \{\boldsymbol{x} \in \Omega \mid d(\boldsymbol{x}, \boldsymbol{p}_i) \leq d(\boldsymbol{x}, \boldsymbol{p}_j) \ \forall \quad i \neq j\} \tag{3.1}$$

where $d(\boldsymbol{x}, \boldsymbol{p}_i)$ is the Euclidean distance $d(\boldsymbol{x}, \boldsymbol{p}_i) = |\boldsymbol{x} - \boldsymbol{p}_i|$ and $i, j = 1, 2, ..., n$. See Figure 3.1 for a standard 2D Voronoi tessellation with 10 seeds. Note that each Voronoi cell contains exactly one seed each.

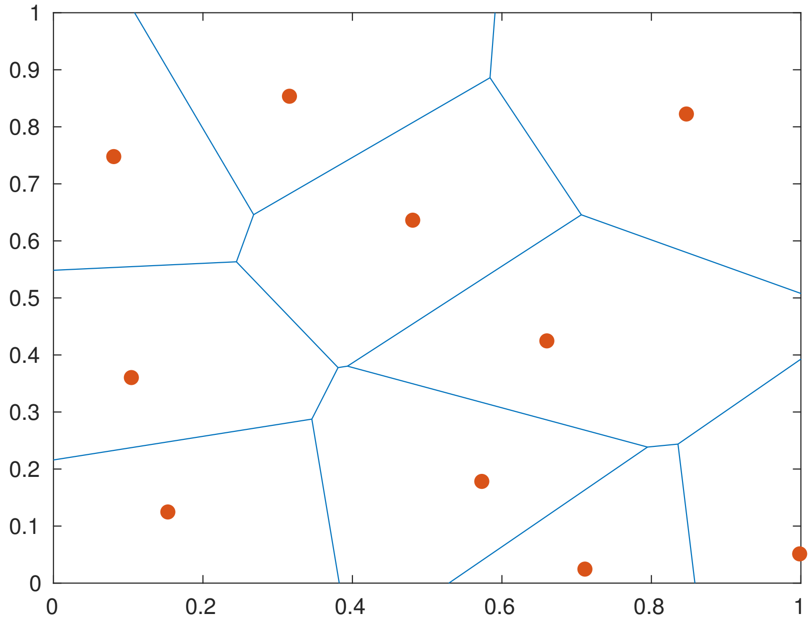


Figure 3.1: Standard 2D Voronoi tessellation with 10 seeds. The blue lines partition the 2D domain, and the orange points are the seeds.

If two seeds are placed very close to each other, then the Voronoi tessellation creates very small cells. Clearly, this is not the case for Figure 3.1 since a minimum distance criterion is implemented, i.e. the seeds have to satisfy:

$$d(\boldsymbol{p}_i, \boldsymbol{p}_j) \geq \frac{R}{2} + \frac{R}{2} \quad \forall\, i \neq j \tag{3.2}$$

where $\frac{R}{2} + \frac{R}{2} = R$ is the specified minimum distance between the seed points. Note that this condition can be visualised as spheres at the position $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, each with radius $\frac{R}{2}$. In order to fulfill this condition the spheres may not overlap.

3.2 Periodic Voronoi tessellation

The issue with the standard Voronoi tessellation is that it causes a clear boundary layer effect. The consequence is that the Voronoi cells close to the boundary are always too large. The boundary layer effect is clearly an artificial consequence caused by the implementation method and must be removed since the aim is to use the Vornoi tessellation to generate a realistic artificial microstructure with as much control as possible. A remedy for this is to introduce the periodic Voronoi tessellation which results in that the Voronoi boundaries become compatible with each other. The implication is that a Voronoi cell that sticks out at one side of the boundary re-enters the Voronoi tessellation from the other side, see Figure 3.2. Note that each Voronoi cell no longer contains exactly one seed each since the empty cells are a part of already existing cells at the opposite boundary.

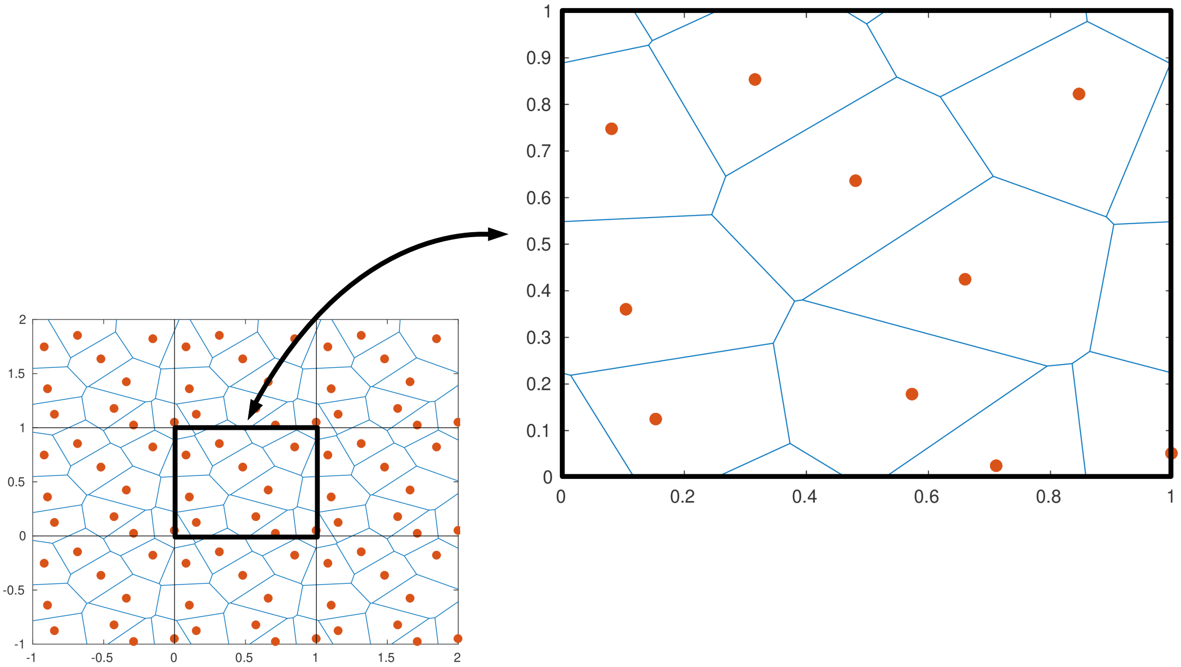


Figure 3.2: Periodic 2D Voronoi tessellation with 10 seeds. Periodicity obtained by extracting the middle 1x1 Voronoi tessellation from the extended 3x3 Voronoi tessellation via truncation.

One issue with the periodic Voronoi tessellation is that it can not respect the minimum distance criterion (3.2) across the boundaries if only one 2D domain is considered. Instead, one has to consider the neighbourhood of the main domain as seen in Figure 3.2; 3x3 domain for 2D and 3x3x3 domain for 3D.

Another benefit of having a periodic Voronoi tessellation is that it lays the foundation for generating a periodic RVE. The artificial SBE microstructure must be an RVE if the theory of computational homogenization is used.

4 Generation of artificial SBE microstructure

One way to create the SBE as a two phase system is by using reaction induced phase separation [4] as described by Ihrner et al. In short, they combined a liquid electrolyte with a stiff vinyl ester based thermoset matrix and let the phase separation be induced during the polymerization. The main advantage is that the combined solution of liquid electrolyte and vinyl ester based thermoset matrix can directly be vacuum-infused onto the carbon fibers and cured in one step. It turns out that the morphology of the SBE varies slightly depending on the exact type of monomer that is combined with the liquid electrolyte. When they combined bisphenol A dimethacrylate ($M = 364.43$ g mol^{-1}), bisphenol A ethoxylate dimethacrylate ($M = 540$ g mol^{-1}), lithium trifluoromethanesulfonate (LiTFS 96%) and 2.2-Dimethoxy-2-phenylacetophenone (DMPA), they produced an SBE with a morphology as shown in Figure 4.1. The figure shows that the SBE is indeed bicontinuous and microporous with connected pore channels. However, it has been difficult to determine more morphological parameters such as the polymer matrix volume fraction of the SBE. Due to lack of data and understanding of the SBE morphology, Figure 4.1 will serve as a rough guideline for the generation of the artificial SBE microstructure.

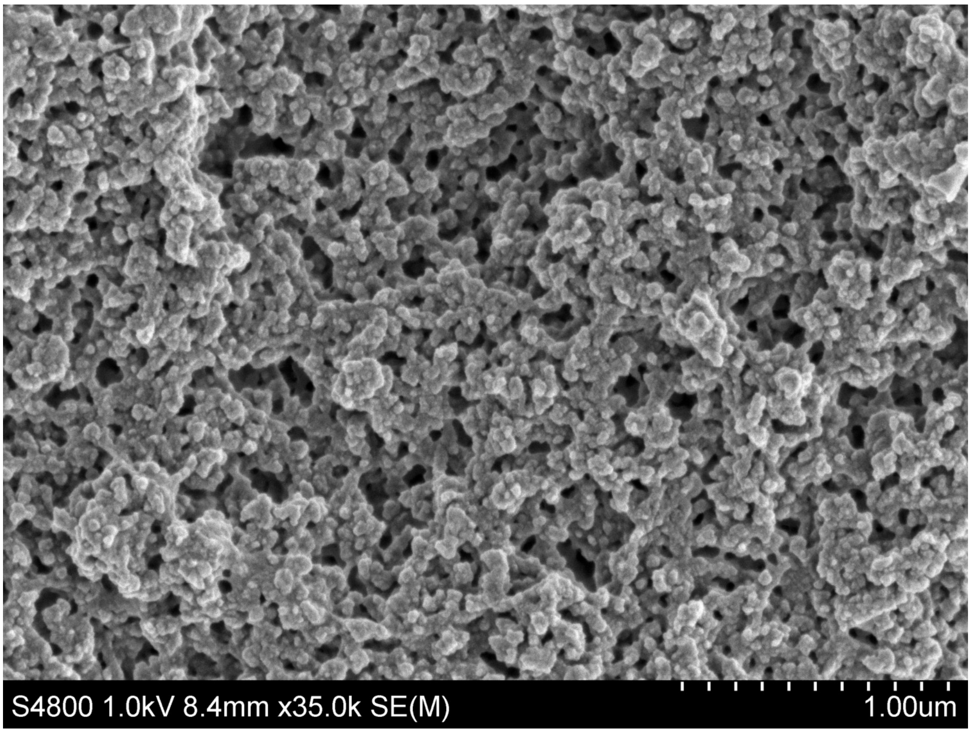


Figure 4.1: Scanning electron microscope (SEM) image of an SBE that reveals the microstructure morphology. Reproduced with permission by Niklas Ihrner [4].

The microstructure generation consists of several steps in different softwares. First the seeds are generated in MATLAB. They are then imported into Voro++ where a periodic Voronoi tessellation is created, which lays the foundation for the artifical microstructure. The Voronoi tessellation goes back to MATLAB again where it is post-processed, and then it is imported into COMSOL. COMSOL generates a new geometry in STL format based on the Voronoi tessellation which represents the SBE, but since the SBE geometry from COMSOL is defective, it is repaired in MATLAB by adding STL triangles to the geometry. Lastly, the geometry is imported into COMSOL one last time where it is converted into a mesh-able solid. The details of each step are presented in the following subsections. An overview of the whole microstructure generation process can be seen in Figure 4.2.

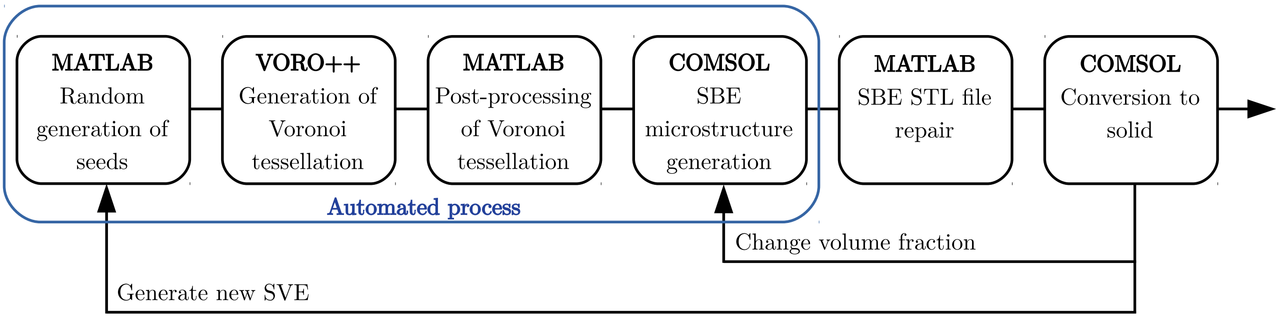


Figure 4.2: Flowchart for the generation of artificial SBE microstructure which shows the software interactions.

4.1 Voronoi tessellation in MATLAB and Voro++

First the seeds need to be placed in order to generate a periodic Voronoi tessellation. This is done in a stochastic fashion based on a uniform distribution inside a unit cube in MATLAB. Since 3 seeds correspond to the simplest artificial microstructure, this case will be used for demonstration throughout the section. However, it is in general preferred to include as many seeds as possible since this increases the amount of statistics in the Voronoi tessellation, which in turn increases the SVE size for the artificial microstructure.

The generation of the periodic Voronoi tessellation itself is done by utilizing Voro++ [17] which is an open-source software library written in C++ that performs 3D computations of Voronoi tessellations. One issue with Voro++ is that the periodic structure that is produced is not contained inside a unit cube, which is a convenient geometric shape to work with when it comes to computational homogenization. The solution is to first obtain the extended structure which can later be truncated to a unit cube, i.e. a 1x1x1 Voronoi tessellation extends to a 3x3x3 Voronoi tessellation where the original Voronoi tessellation is in the middle. This post-processing step is performed in MATLAB. Note that only the Voronoi cells that are partially inside the unit cube are of interest. The remaining cells are filtered away, see Figure 4.3 and 4.4.

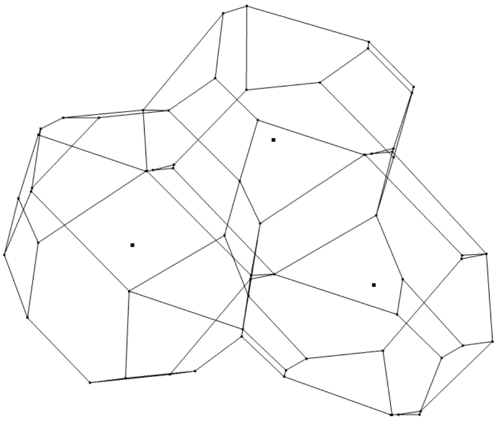


Figure 4.3: Voronoi tessellation with 3 seeds in 3D.

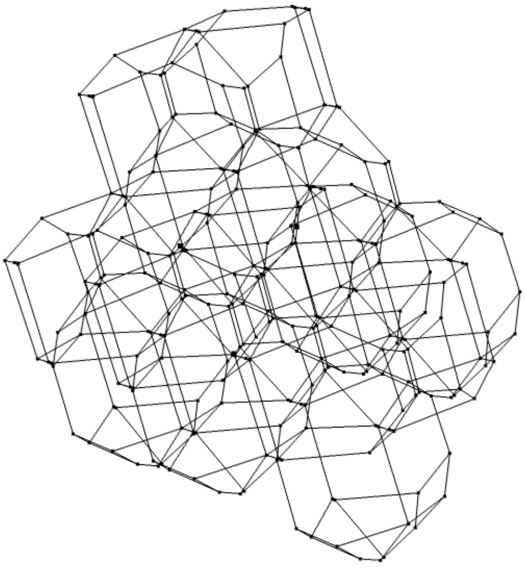


Figure 4.4: Extended Voronoi tessellation with 3 seeds in 3D.

4.2 Heat manipulation in COMSOL Multiphysics

The next step is to import the extended Voronoi tessellation as points (seeds) and lines (cell edges) into COMSOL where it is truncated such that only the parts that are inside the unit cube are preserved, see Figure 4.5.

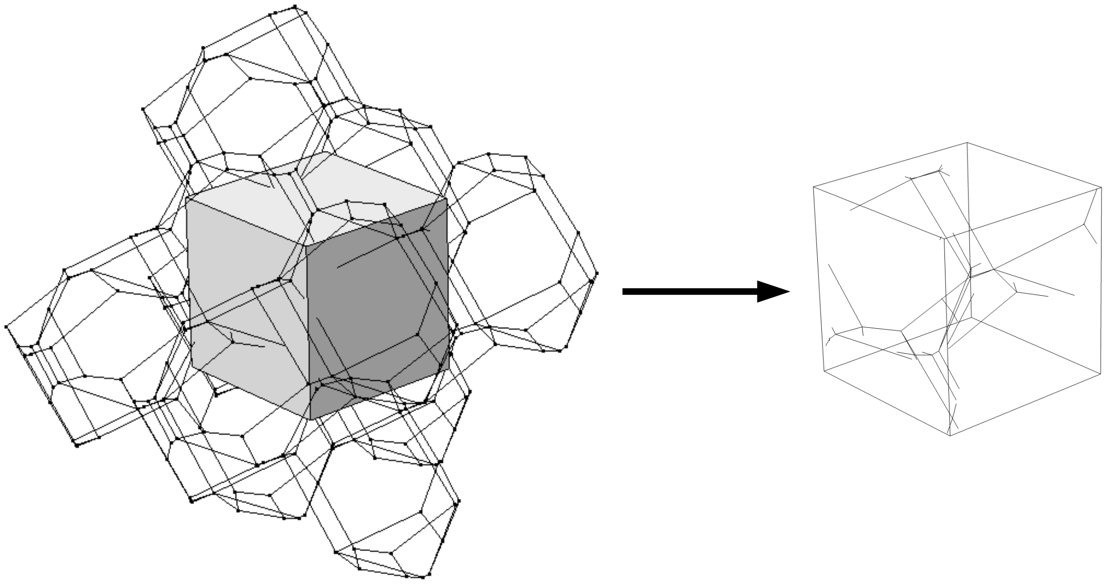


Figure 4.5: Truncation of extended Voronoi tessellation to obtain a periodic structure contained in a unit cube.

The result is a periodic Voronoi tessellation that is embedded in a solid cube. It turns out that by assigning a unit thermal conductivity to the solid cube, and choosing heat sinks and heat sources in a clever fashion, it becomes possible to manipulate heat in order to generate a geometry that resembles the SBE. By solving the heat equation with SPBC where the seeds are heat sinks, and the remaining parts of the Voronoi tessellation are heat sources, an isosurface and isovolume that resembles the SBE, as shown in Figure 4.1, can be obtained. The isovolume is obtained by applying an element filter on the mesh that reveals all elements with nodal values that are above a certain temperature, see Figure 4.6 and 4.7. Note that the number of seeds that are used in the Voronoi tessellation will roughly correspond to the number of pores that will appear in the artificial microstructure. Moreover, since the heat equation was solved a fully periodic fashion, the resulting isosurface and isovolume also becomes periodic.

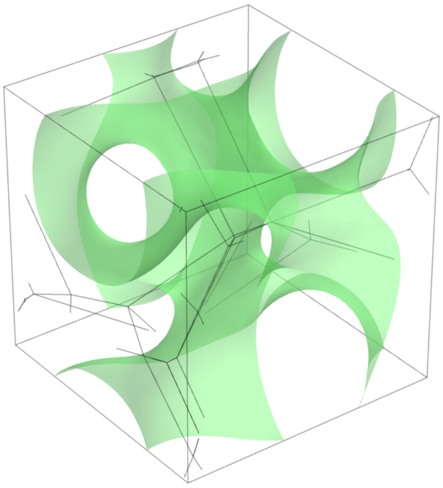


Figure 4.6: Isosurface based on 3 seeds.

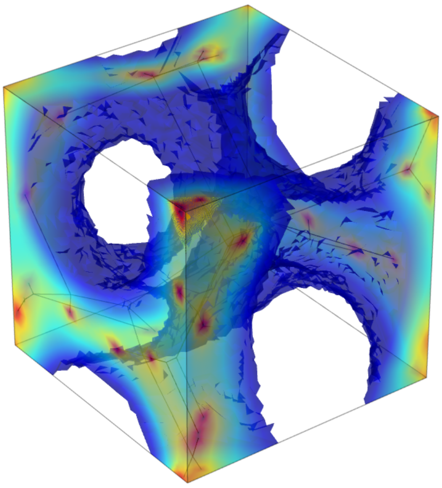


Figure 4.7: Isovolume based on 3 seeds.

It is possible to obtain different polymer matrix volume fractions of the structure by adjusting the isolevel. However, one issue is that the continuous pore channels might close and become isolated at high volume fractions. When this happens, the microstructure is no longer bicontinuous. It turns out that this method can not generate artificial bicontinuous microstructures above around 70% volume fraction.

Both the isosurface and the isovolume can be extracted from COMSOL as STL files. Since the isosurface is just a surface made up by STL triangles, it needs to be post-processed in MATLAB order to form a closed solid. Lastly, note that combining the isosurface and the isovolume gives a microstructure that resembles a closed solid, see Figure 4.8 - 4.11.

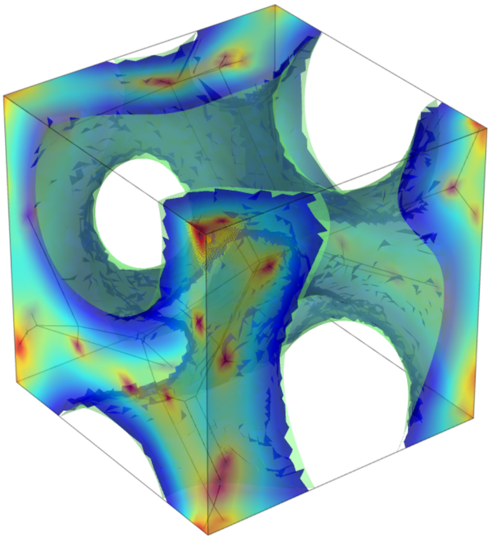


Figure 4.8: Combination of isosurface and isovolume based on 3 seeds with 48.02% polymer matrix volume fraction.

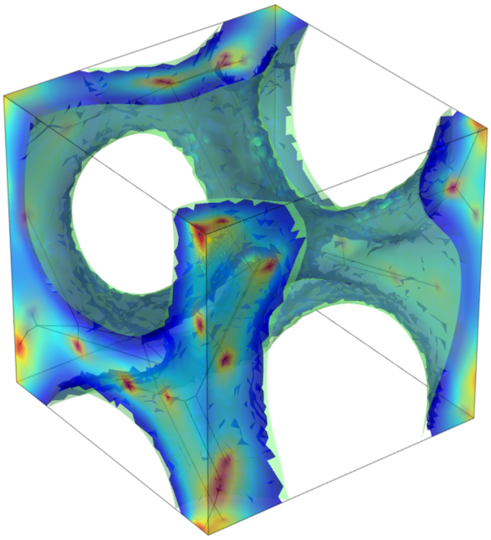


Figure 4.9: Combination of isosurface and isovolume based on 3 seeds with 30.28% polymer matrix volume fraction.

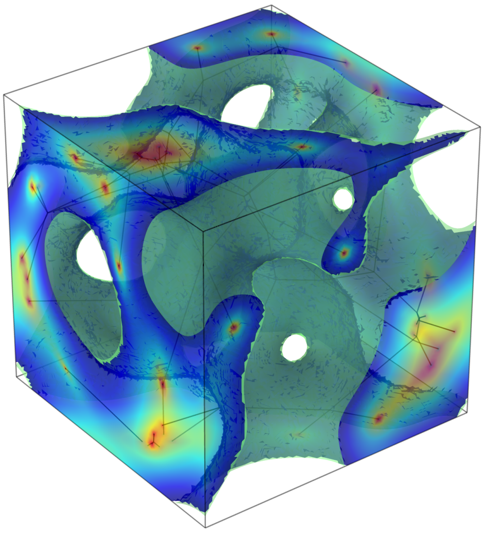


Figure 4.10: Combination of isosurface and isovolume based on 10 seeds with 46.44% polymer matrix volume fraction.

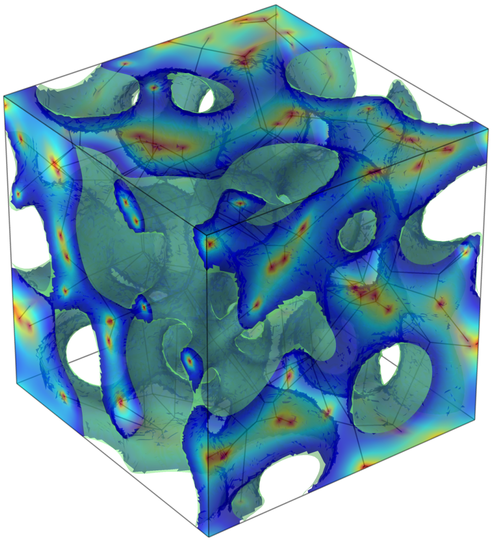


Figure 4.11: Combination of isosurface and isovolume based on 20 seeds with 39.31% polymer matrix volume fraction.

4.3 Combining and repairing the isosurface and isovolume

The isovolume in itself already corresponds to a closed solid geometry, but it can not be used directly as the artificial microstructure since the surface roughness is too high. In fact, each element is clearly visible from the element filtering in COMSOL. The coarse surface of the isovolume would lead to high computational costs, and it would affect the meshing of the geometry since it is already discretized. Hence, it is of interest to use the isosurface as a geometric shell for the artificial microstructure due to the smoothness of the surface. The smoothness of the isosurface is a result of COMSOL's interpolation algorithm between the nodal values. In order for the isosurface to become an airtight solid, the openings at the cube faces of the isosurface need to be patched up. This post-processing step is performed in MATLAB.

The STL files of the isosurface and isovolume are imported into MATLAB. STL files are surface representations of geometries, where the surface is made up by a network of connected triangles. Each STL triangle consists of 3 vertices and 1 outward-pointing normal vector, see Figure 4.12.

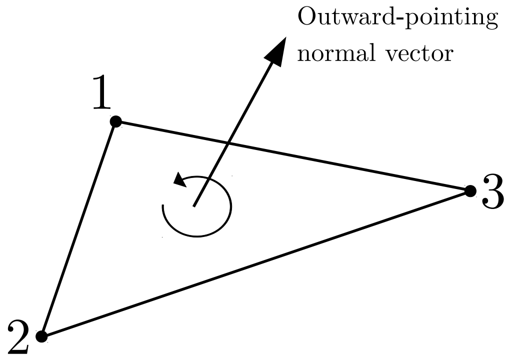


Figure 4.12: The constituents of an STL triangle.

For the post-processing step, only the triangle vertices will be needed. Note that the interior vertices of the isosurface and isovolume are not of interest. Since the goal is to form a closed solid out of an open isosurface geometry, only the vertices at the 6 cube faces are considered, see Figure 4.13 and 4.14. Furthermore, the data is split such that only one cube face is treated at the time, see Figure 4.15 and 4.16.

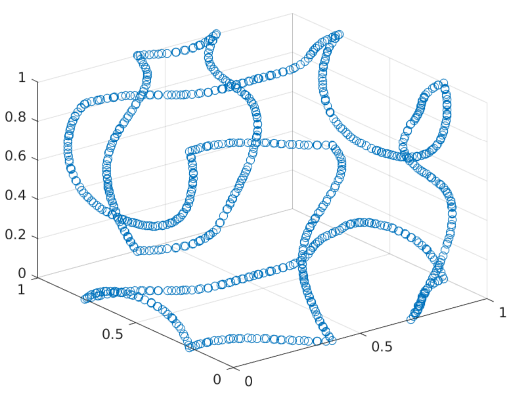


Figure 4.13: Imported boundary vertices from isosurface STL in 3D.

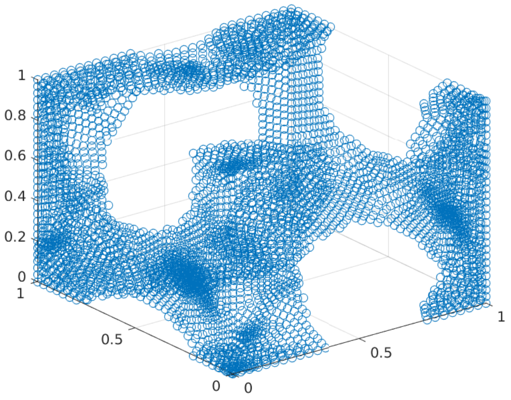


Figure 4.14: Imported boundary vertices from isovolume STL in 3D.

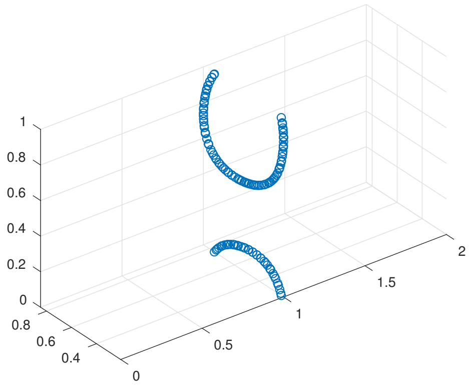


Figure 4.15: Isosurface vertices on image cube face in x-direction.

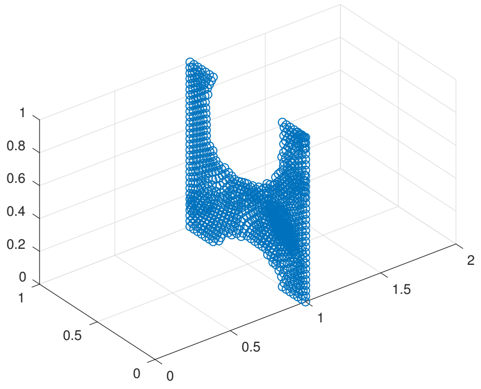


Figure 4.16: Isovolume vertices on image cube face in x-direction.

As shown in Figure 4.6, there are large holes at the cube faces of the isosurface. These must be patched up in order to obtain an airtight solid. A way to achieve this is to modify the original isosurface STL file and manually add more triangles such that the isosurface results in a volume. By combining the data from the isosurface and the isovolume in Figure 4.15 and 4.16, a poor preliminary Delaunay triangulation can be performed, see Figure 4.17. The Delaunay triangulation is a method to generate a network of connected triangles based on a given point cloud. This triangulation method tries to maximize the minimum angle such that the triangles obtain a good aspect ratio.

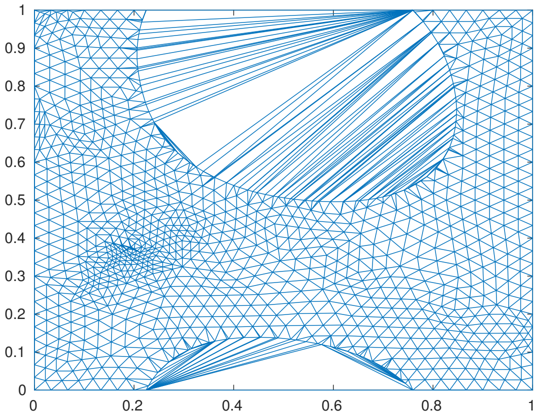


Figure 4.17: Preliminary Delaunay triangulation to be used for patching up the holes on the isosurface.

The reason Figure 4.17 is a preliminary Delaunay triangulation is because the Delaunay triangulation indiscriminately generates triangles everywhere, which is an unwanted effect in this case since some parts of the microstructure should be the polymer matrix and the rest should be the pores. The small triangles with good aspect ratio in Figure 4.17 correspond to the polymer matrix, and the two semi-ellipses at the top and bottom correspond to the pores which should not be triangulated. An algorithm was developed in order to remove the triangles that were generated inside the pores. Note that it is necessary to identify the pore-polygons which in this case correspond to the two semi-ellipses. In short, the task of the algorithm is to perform polygon tracing of the isosurface vertices, see Algorithm 1.

```
Initialize MATLAB
    Load the isosurface vertices per cube face;
    Pick any vertex as the first vertex;
    Pick the second vertex, which is the vertex that is closest to the first one;
    Save these two initial vertices in a vertex list;
    while there exist unidentified vertices in the neighbourhood do
        Define a search direction based on the 2 latest vertices from the vertex list;
        Define a normal vector associated with the STL triangle orientation for the latest vertex from the
          vertex list;
        Pick the closest point that has a small search direction deviation and a small normal vector deviation;
        if closest point is closer than user-defined minimum distance then
            Add the closest point to the vertex list;
        else if special case at edge or corner then
            Apply special rule that ignores the user-defined minimum distance;
            Add the best available candidate to the vertex list;
        else
            Exit while loop since there are no more candidates in the neighbourhood;
        end
    end
End MATLAB
```

Algorithm 1: Simplified overview of the polygon tracing algorithm for distinguishing the polymer matrix from the pores.

By using Algorithm 1 it is possible to identify and separate each individual polygon from Figure 4.15, see Figure 4.18 and 4.19.

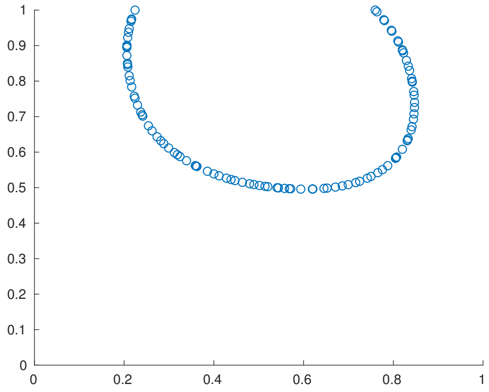


Figure 4.18: Identifying the top semi-ellipse with the polygon tracing algorithm.

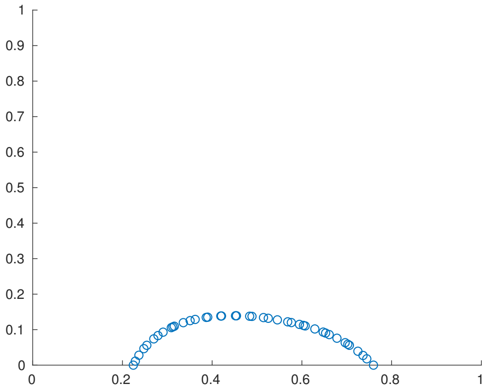


Figure 4.19: Identifying the bottom semi-ellipse with the polygon tracing algorithm.

One way to filter out the unwanted mesh triangles from the preliminary Delaunay triangulation is to consider whether or not the mesh triangle midpoint is inside any polygons. If a mesh triangle midpoint is inside any of the identified polygons, then that is an unwanted mesh triangle which should be removed. See the final filtered Delaunay triangulation in Figure 4.20.

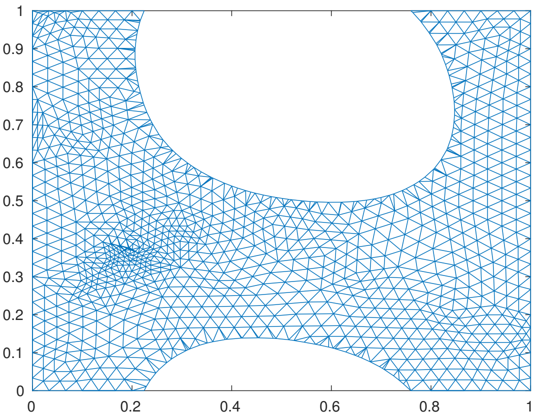


Figure 4.20: Final filtered Delaunay triangulation to be used for patching up the holes on the isosurface.

After completing the post-processing of the STL data, the Delaunay triangulation is finally added to the isosurface STL file. Doing this 5 additional times for the remaining cube faces results in a patched up isosurface which encloses a volume. The last step is to import the modified isosurface STL file into COMSOL where the STL file is repaired and converted into a solid, see Figure 4.21.

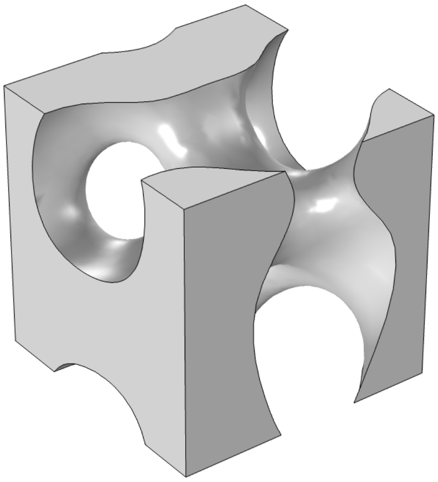


Figure 4.21: Modified isosurface STL file that has been repaired and converted into a solid in COMSOL.

4.4 Issues with the artificial SBE microstructure generation

As shown in Figure 4.2, a large part of the microstructure generation process is automated. Although the seed generation is performed in MATLAB, executing operating system commands in MATLAB makes it possible to directly use MATLAB's output to create the Voronoi tessellation in Voro++. The output from Voro++ can then be seamlessly imported into MATLAB and also post-processed. The next step is to import the Voronoi tessellation into COMSOL and perform the heat manipulation, which is a step that can also be fully controlled from MATLAB by using the COMSOL LiveLink module. In short, generating the structures shown in Figure 4.8 - 4.11 from the very beginning requires minimal effort since the whole process is automated and performed by one single MATLAB script.

Despite this, mass generation of artificial SBE microstructure is still a tall task since selecting a suitable volume fraction for the microstructure is difficult. Not all STL files can be patched up and repaired, especially if they contain several small details and defects. This means that before extracting the STL data from COMSOL, the user has to carefully inspect whether or not it is viable to repair it. Additionally, it seems that the difficulty in obtaining repairable STL files increases for increasing number of seeds in the Voronoi tessellation. This effectively means that the artificial microstructure can only have a limited number of seeds, and that the volume fraction is not chosen freely.

After trial and error, it turns out that the highest number of seeds that consistently results in viable artificial microstructures is 10. While 3 seeds are used to demonstrate the microstructure generation process, 10 seeds are used for the simulations in this project. Furthermore, it was found that microstructures with low volume fractions constantly cause issues. The volume fraction range for the polymer matrix which consistently results in viable artificial microstructures corresponds to around 45% - 70%.

4.5 The inverse artificial SBE microstructure

A remedy to the above-mentioned constraint in volume fraction range is to introduce the inverse artificial SBE microstructure. It is possible to perform boolean operators in COMSOL which can result in the inverse of the microstructure in Figure 4.21. Although the reference microstructure that is being mimicked is Figure 4.1, it is still not completely clear whether the original microstructure or the inverse microstructure is more similar to the real SBE microstructure. The only definitive constraints that the artificial microstructure must fulfill are the properties of micro-porosity and bicontinuity, which are properties that the inverse microstructure also has, see Figure 4.22.

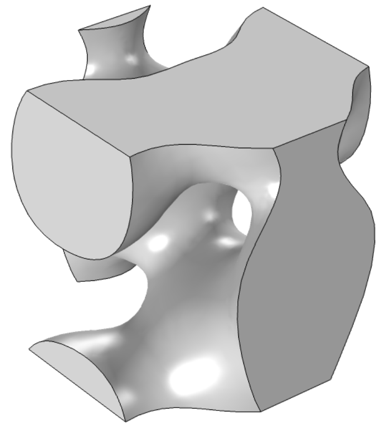


Figure 4.22: Inverted microstructure with 3 seeds by performing boolean operators in COMSOL.

The consequence of introducing the inverse artificial SBE microstructure is that the solid polymer phase and the pore space swap places. The main advantage of this method is that the new viable volume fraction range is 30% - 70%. This is a more realistic range since a volume fraction above 70% will probably have too much of the solid polymer matrix which results in great mechanical properties, but bad ion conductivity. The reverse will also be true, i.e. a volume fraction below 30% probably results in poor mechanical properties but good ion conductivity.

4.6 Mass generation of artificial SBE microstructures with 10 seeds

Due to the limitation on the maximum number of seeds per Voronoi tessellation, the artificial microstructure is more of an SVE rather than an RVE. The evaluation of the effective properties will become slightly different for every realization of the artificial microstructure. Since the scatter of the effective properties depend on how much statistics that are included in the SVE, a robust strategy (statistical ensembling) is to first generate several SVE realizations at the same volume fraction and then compute the average effective properties based on a large enough sample.

As mentioned above, the volume fraction is not chosen freely. Hence, it is difficult to generate many SVE realizations with the exact same volume fraction. For simplicity, a generous ±1.5% volume fraction tolerance is introduced, where e.g. everything between 70% and 67% will be considered to have the volume fraction 68.5%. In preparation for the simulations, 46 SBE microstructures were generated. Unfortunately, some of the structures failed the meshing stage or had convergence issues, which explains why some cases have more structures than others. The volume fraction distribution of the generated SBE microstructures is presented in Table 4.1:

Table 4.1: Volume fraction (VF) groups of original and inverse structures.

Number of structures per VF group				
		Elasticity: Drained	**Elasticity: Undrained**	**Chemical diffusion**
Original structure	VF 68.5%:	5 out of 7	3 out of 7	6 out of 7
	VF 65.5%:	3 out of 5	3 out of 5	3 out of 5
	VF 62.5%:	6 out of 7	7 out of 7	6 out of 7
	VF 59.5%:	3 out of 4	3 out of 4	3 out of 4
Inverse structure	VF 31.5%:	5 out of 7	3 out of 7	6 out of 7
	VF 34.5%:	3 out of 5	3 out of 5	3 out of 5
	VF 37.5%:	6 out of 7	7 out of 7	6 out of 7
	VF 40.5%:	3 out of 4	3 out of 4	3 out of 4
	Total:	**34 out of 46**	**32 out of 46**	**36 out of 46**

For demonstration purpose, the successful SBE microstructures in the volume fraction group 65.5% and 34.5% are presented, see Figure 4.23-4.28.

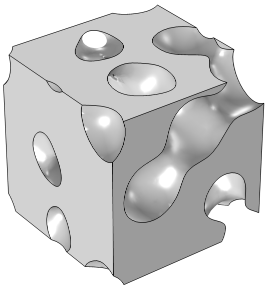


Figure 4.23: Original structure VF 66.17%.

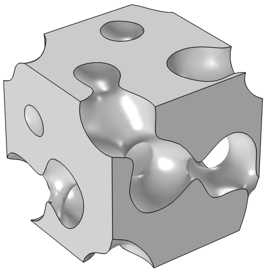


Figure 4.24: Original structure VF 65.10%.

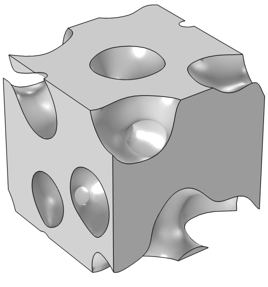


Figure 4.25: Original structure VF 64.16%.

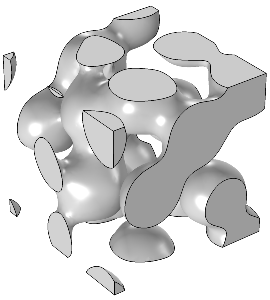


Figure 4.26: Inverse structure VF 33.83%.

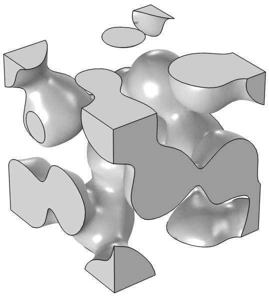


Figure 4.27: Inverse structure VF 34.90%.

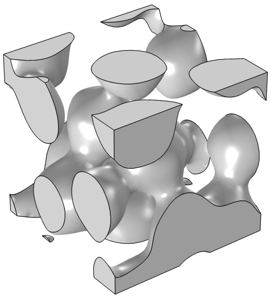


Figure 4.28: Inverse structure VF 35.84%.

5 Mesh convergence analysis

Before evaluating the multifunctional performance of the SBE, a convergence study is performed to serve as a rough guideline for which mesh quality to use. The assessment of mesh convergence can be performed by investigating some scalar quantity which can serve as a convergence indicator. One such example is to observe the largest eigenvalue λ_1 of the constitutive matrix. For simplicity, the convergence study is only done for one artificial SBE microstructure, which in this case is one of the structures from the group VF 31.5%. It is assumed that the converged mesh for one artificial SBE microstructure will roughly be the same as for the other structures. Another interesting convergence indicator is to compute the homogenized strain energy for linear elasticity, and free energy for stationary diffusion:

$$\tilde{\Psi}_i = \langle \boldsymbol{\epsilon}_i \cdot \mathbf{E} \cdot \boldsymbol{\epsilon}_i \rangle_\square \quad \text{(Drained elasticity)} \tag{5.1}$$

$$\hat{\Psi}_i = \langle (\boldsymbol{\nabla} u)_i \cdot \boldsymbol{D} \cdot (\boldsymbol{\nabla} u)_i \rangle_\square \quad \text{(Diffusion)} \tag{5.2}$$

where $i = \{x, y, z\}$ indicates the direction of the macroscopic driving force. Furthermore, the computed convergence indicators are normalized with respect to a converged overkill solution:

$$\begin{aligned}
\text{Normalized } \tilde{\lambda}_1\text{: } & \frac{\tilde{\lambda}_1}{\tilde{\lambda}_{1,\infty}} \quad \text{(Drained elasticity)} \\
\text{Normalized } \hat{\lambda}_1\text{: } & \frac{\hat{\lambda}_1}{\hat{\lambda}_{1,\infty}} \quad \text{(Diffusion)} \\
\text{Normalized } \tilde{\Psi}_1\text{: } & \frac{\tilde{\Psi}_i}{\tilde{\Psi}_{i,\infty}} \quad \text{(Drained elasticity)} \\
\text{Normalized } \hat{\Psi}_1\text{: } & \frac{\hat{\Psi}_i}{\hat{\Psi}_{i,\infty}} \quad \text{(Diffusion)}
\end{aligned}$$

It should be noted that the volume averaged convergence indicators such as Equation (5.1) and (5.2) converge even for relatively coarse meshes. The reason is that they are homogenized global quantities that smear out the effects of local nonconvergence. The homogenized free energies are suitable convergence indicators since the quantities of interest are the homogenized effective properties.

5.1 Mesh choice for linear elasticity

For simplicity, the convergence study is only done for the drained case since the undrained case requires a more complex implementation. The meshing is performed in COMSOL's native meshing environment, and the user-defined parameters that are investigated in the convergence study are presented in Table 5.1.

Table 5.1: Intervals of COMSOL's user-defined mesh parameters that are used in the convergence test.

User-defined mesh parameters	
Number of DOF:	[17.7, 314.4] $\cdot 10^3$
Number of elements:	[2.617, 65.64] $\cdot 10^3$
Maximum element size:	[0.508, 0.045]
Minimum element size:	[0.071, 0.006]
Maximum element growth rate:	[2.00, 1.46]
Curvature factor:	[1.00, 0.505]
Resolution of narrow regions:	[0.100, 0.595]

Figure 5.1 shows how the largest eigenvalue converges at around 100000 DOF. Figure 5.2 shows that the energy indicator $\tilde{\Psi}_i$ from Equation 5.1 also converges at around 100000 DOF. While the largest eigenvalue $\tilde{\lambda}_1$ from the

elasticity matrix seems to converge in a stable manner, the strain energies $\tilde{\Psi}_i$ have some fluctuations early on. However, as soon as the number of DOF surpasses 100000, the strain energies $\tilde{\Psi}_i$ stabilizes and converges.

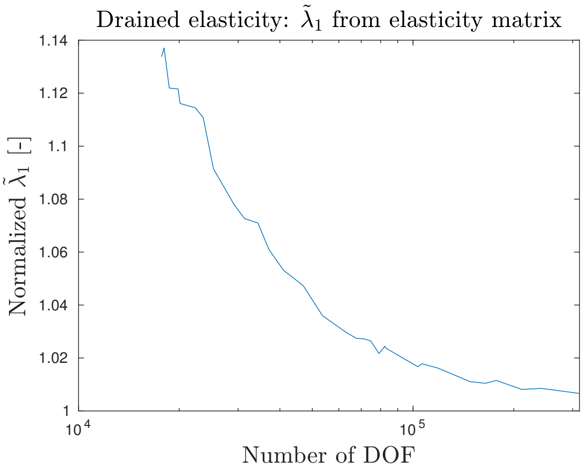


Figure 5.1: Normalized largest eigenvalue from homogenized elasticity matrix.

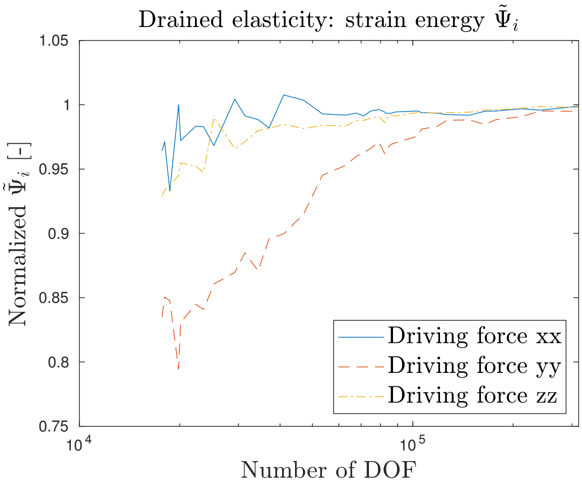


Figure 5.2: Normalized strain energy with a macroscopic driving force in different directions.

The conclusion of the convergence study for linear elasticity is to use the user-defined mesh settings as shown in Table 5.2:

Table 5.2: Converged mesh parameters in COMSOL.

User-defined mesh parameters	
Number of DOF:	103434
Number of elements:	15975
Maximum element size:	0.08762
Minimum element size:	0.01173
Maximum element growth rate:	1.50528
Curvature factor:	0.55025
Resolution of narrow regions:	0.54975

5.2 Mesh choice for stationary diffusion

The investigated user-defined mesh parameters for stationary diffusion are presented in Table 5.3.

Table 5.3: Intervals of COMSOL's user-defined mesh parameters that are used in the convergence test.

User-defined mesh parameters	
Number of DOF:	[5.90, 104.4] $\cdot 10^3$
Number of elements:	[2.617, 65.35] $\cdot 10^3$
Maximum element size:	[0.508, 0.045]
Minimum element size:	[0.071, 0.006]
Maximum element growth rate:	[2.00, 1.46]
Curvature factor:	[1.00, 0.505]
Resolution of narrow regions:	[0.100, 0.595]

Figure 5.3 shows how the largest eigenvalue converges at around 40000 DOF. Figure 5.4 shows how the energy indicator $\hat{\Psi}_i$ from Equation 5.2 also converges at around 40000 DOF.

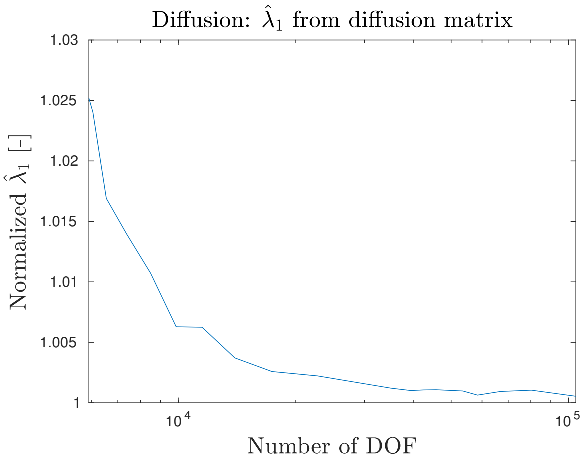


Figure 5.3: Normalized largest eigenvalue from homogenized diffusion coefficient matrix.

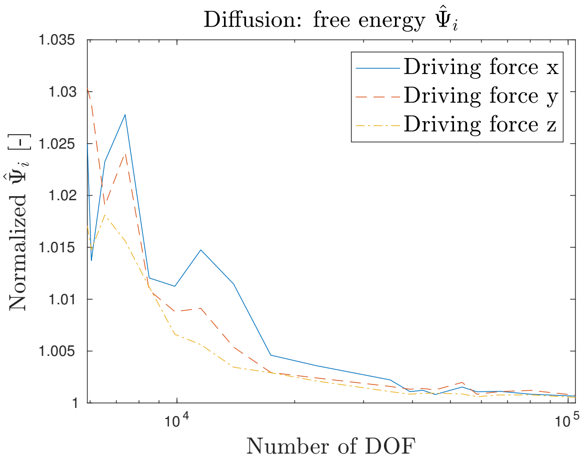


Figure 5.4: Normalized free energy with a macroscopic driving force in each direction.

The conclusion of the convergence study for stationary diffusion is to use the user-defined mesh settings as shown in Table 5.4:

Table 5.4: Converged mesh parameters in COMSOL.

User-defined mesh parameters	
Number of DOF:	39422
Number of elements:	21785
Maximum element size:	0.0811
Minimum element size:	0.0101
Maximum element growth rate:	1.45
Curvature factor:	0.500
Resolution of narrow regions:	0.600

6 Virtual material testing

The effective properties are obtained by following the procedure described in Figure 2.1 and 2.4. The sub-scale material parameters for linear elasticity were set to $E = 1$ (dimensionless) and $\nu = 0.499$ since the polymer matrix is assumed to be close to incompressibility. These choices result in an effective stiffness matrix that is normalized with respect to the Young's modulus. A similar approach was used for the diffusion problem where the sub-scale diffusion coefficient was set to $D = 1$ (dimensionless). Yet again, this results in a normalized effective diffusion coefficient matrix.

6.1 Effective stiffness of SBE

The COMSOL implementation of the isotropic sub-scale constitutive relationship is based on the Voigt format:

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{bmatrix} \tag{6.1}$$

The implementation of the macro-scale constitutive relationship is also in Voigt format:

$$\begin{bmatrix} \bar{\sigma}_{11} \\ \bar{\sigma}_{22} \\ \bar{\sigma}_{33} \\ \bar{\sigma}_{23} \\ \bar{\sigma}_{13} \\ \bar{\sigma}_{12} \end{bmatrix} = \begin{bmatrix} \bar{E}_{1111} & \bar{E}_{1122} & \bar{E}_{1133} & \bar{E}_{1123} & \bar{E}_{1113} & \bar{E}_{1112} \\ \bar{E}_{2211} & \bar{E}_{2222} & \bar{E}_{2233} & \bar{E}_{2223} & \bar{E}_{2213} & \bar{E}_{2212} \\ \bar{E}_{3311} & \bar{E}_{3322} & \bar{E}_{3333} & \bar{E}_{3323} & \bar{E}_{3313} & \bar{E}_{3312} \\ \bar{E}_{2311} & \bar{E}_{2322} & \bar{E}_{2333} & \bar{E}_{2323} & \bar{E}_{2313} & \bar{E}_{2312} \\ \bar{E}_{1311} & \bar{E}_{1322} & \bar{E}_{1333} & \bar{E}_{1323} & \bar{E}_{1313} & \bar{E}_{1312} \\ \bar{E}_{1211} & \bar{E}_{1222} & \bar{E}_{1233} & \bar{E}_{1223} & \bar{E}_{1213} & \bar{E}_{1212} \end{bmatrix} \begin{bmatrix} \bar{\epsilon}_{11} \\ \bar{\varepsilon}_{22} \\ \bar{\varepsilon}_{33} \\ 2\bar{\varepsilon}_{23} \\ 2\bar{\varepsilon}_{13} \\ 2\bar{\varepsilon}_{12} \end{bmatrix} \tag{6.2}$$

The computation of the effective stiffness is done by setting the macro-scale strains to unit strains componentwise. Consider the unit strain $\bar{\epsilon}_{11} = 1$ in 11-direction:

$$\underline{\boldsymbol{\epsilon}}_{11} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.3}$$

Matrix multiplication with the stiffness matrix from (6.2) gives:

$$\begin{bmatrix} \bar{\sigma}_{11} \\ \bar{\sigma}_{22} \\ \bar{\sigma}_{33} \\ \bar{\sigma}_{23} \\ \bar{\sigma}_{13} \\ \bar{\sigma}_{12} \end{bmatrix} = \begin{bmatrix} \bar{E}_{1111} \\ \bar{E}_{2211} \\ \bar{E}_{3311} \\ \bar{E}_{2311} \\ \bar{E}_{1311} \\ \bar{E}_{1211} \end{bmatrix} \tag{6.4}$$

which means that the stiffness matrix is simply computed columnwise and obtained by volume averaging the sub-scale stress to macro-scale stress $\bar{\boldsymbol{\sigma}} = \langle \boldsymbol{\sigma} \rangle_{\square}$ in a post-processing step. The same procedure is repeated for all six strain components. Note that for the shear strain component in 12-direction, the unit strain corresponds to $\bar{\epsilon}_{12} = \frac{1}{2}$. This is true for all the shear strain components since they exist in symmetric pairs.

Figure 6.1-6.3 illustrates how the SBE microstructure is deformed by the componentwise input macro-strain $\bar{\epsilon}_{ij}$ during the virtual material testing. The deformations in the figures correspond to the total sub-scale deformation $\boldsymbol{u} = \boldsymbol{u}^{\mathrm{M}} + \boldsymbol{u}^{\mathrm{S}}$. The deformations are obtained by solving the drained linear elasticity problem on an example structure (VF 63.06 % polymer matrix) using the unit strain $\bar{\epsilon}_{11} = 1$ and unit shear strain $\bar{\epsilon}_{12} = \frac{1}{2}$.

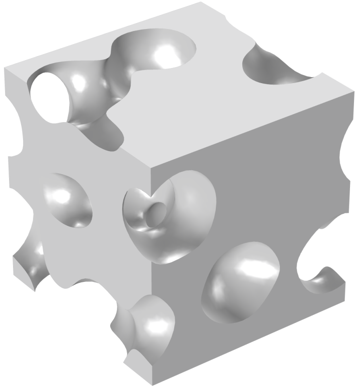


Figure 6.1: Undeformed body of polymer matrix.

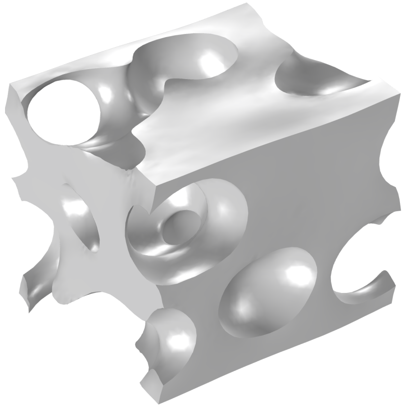


Figure 6.2: Deformed body by normal strain $\bar{\epsilon}_{11} = 1$.

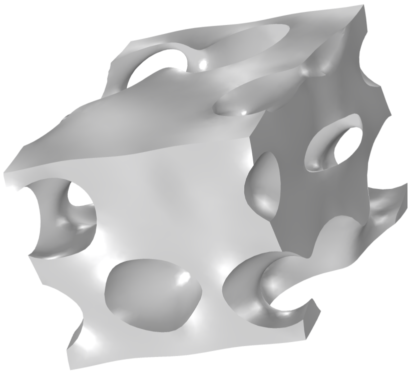


Figure 6.3: Deformed body by shear strain $\bar{\epsilon}_{12} = \frac{1}{2}$.

The corresponding fluctuation field $\boldsymbol{u}^{\mathrm{S}}$ from the unit normal strain $\bar{\epsilon}_{11} = 1$ is given in Figure 6.4-6.6.

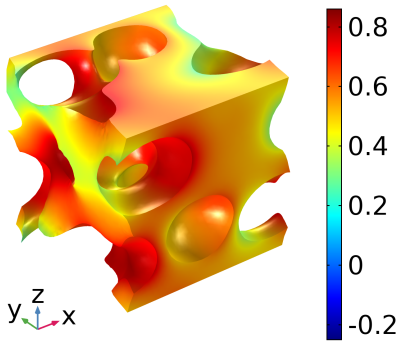


Figure 6.4: Fluctuation field u_x^{S} in x-direction by $\bar{\epsilon}_{11} = 1$.

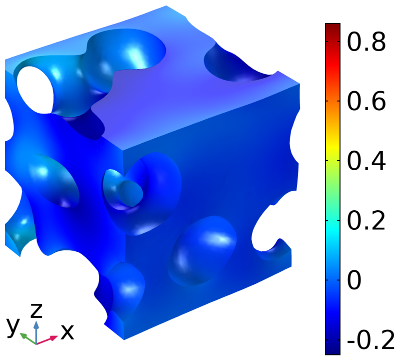


Figure 6.5: Fluctuation field u_y^{S} in y-direction by $\bar{\epsilon}_{11} = 1$.

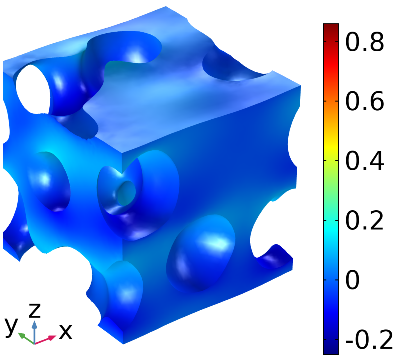


Figure 6.6: Fluctuation field u_z^{S} in z-direction by $\bar{\epsilon}_{11} = 1$.

The corresponding von Mises effective stress σ^{vM} from the unit normal strain $\bar{\epsilon}_{11} = 1$ is given in Figure 6.7-6.9.

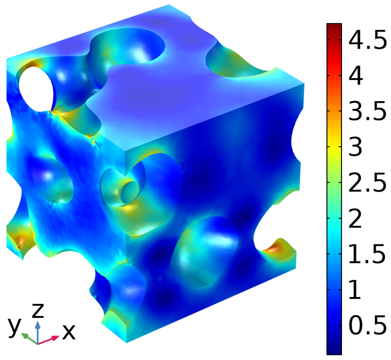


Figure 6.7: Volume plot of von Mises effective stress σ^{vM}.

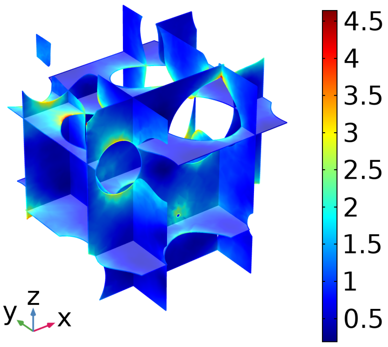


Figure 6.8: Multislice plot of von Mises effective stress σ^{vM}.

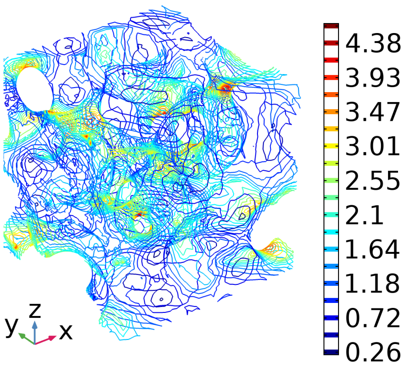


Figure 6.9: Contour plot of von Mises effective stress σ^{vM}.

After solving both the drained and undrained sub-scale problem, the elasticity matrix in Voigt format is obtained for respective case. Figure 6.10 and 6.12 reveal the relationship between the structural stiffness and the volume fraction of polymer matrix. The curve shows the ensemble average of the normalized largest eigenvalues of each VF group, while the vertical bars show the spread. Each vertical bar corresponds to the range $\mu \pm \sigma$ where μ is the average and σ is the standard deviation.

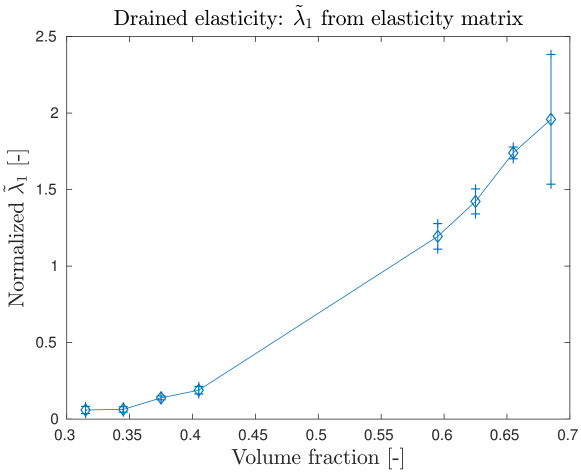


Figure 6.10: $\tilde{\lambda}_1$ in drained case. In total 8 statistical ensembles based on 34 microstructure realizations.

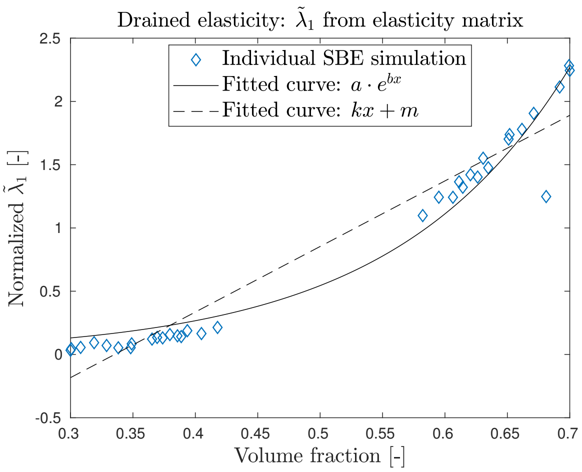


Figure 6.11: $\tilde{\lambda}_1$ in drained case. Individual SBE simulations without VF grouping.

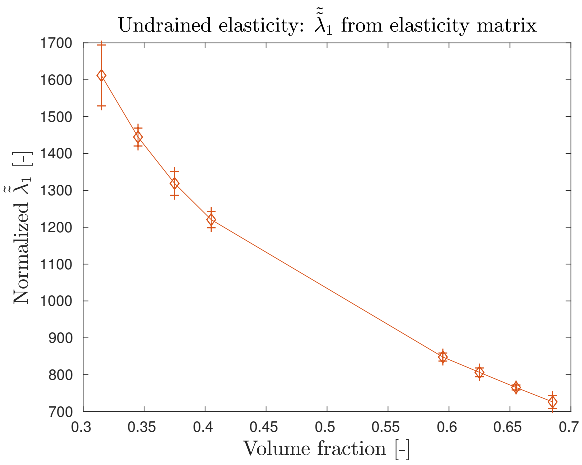


Figure 6.12: $\tilde{\tilde{\lambda}}_1$ in undrained case. In total 8 statistical ensembles based on 32 microstructure realizations.

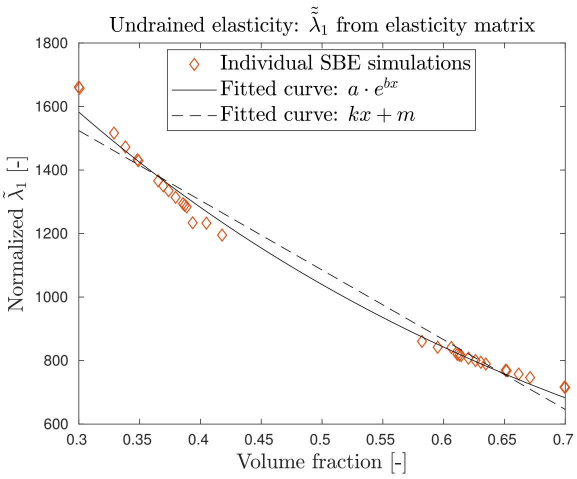


Figure 6.13: $\tilde{\tilde{\lambda}}_1$ in undrained case. Individual SBE simulations without VF grouping.

Combining both plots gives Figure 6.14 which shows that the undrained case is the upper bound while the drained case is the lower bound for the SBE's structural stiffness.

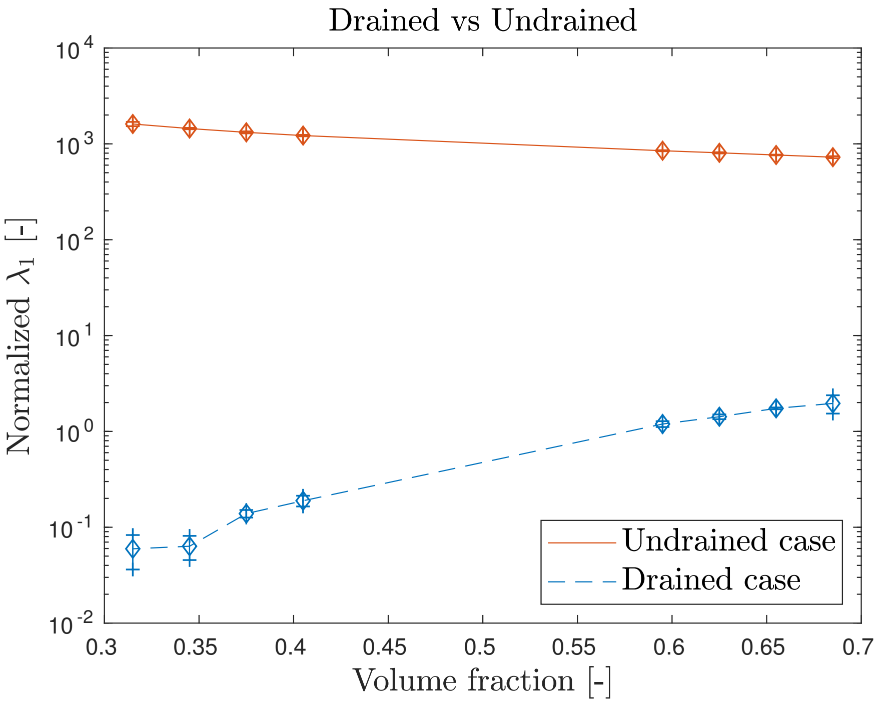


Figure 6.14: Upper bound (undrained) and lower bound (drained) of λ_1 in a semi-log plot.

It seems that the curves in both cases show some non-linearity in the range of volume fraction that is studied. It can also be noted that the structural stiffness increases for increasing volume fraction of polymer matrix in the drained case, which is as expected. It may seem unintuitive that the structural stiffness decreases for increasing volume fraction in the undrained case, but this is in fact a consequence of the extreme assumptions that are used to compute the upper bound. Since the liquid electrolyte is incompressible, it has a much larger bulk modulus than the polymer matrix. Thus, the liquid electrolyte contributes more to the largest eigenvalue than the polymer matrix does. This can only happen when the liquid electrolyte is completely trapped by a very stiff membrane. However, in reality the membrane will not be stiff enough to fully trap the liquid electrolyte inside the pores. See Figure 6.15-6.18 to consider the normal component $\bar{E}_{1111}$ and shear component $\bar{E}_{1212}$ separately. Note that the shear components are almost identical, the differences are found in the normal components.

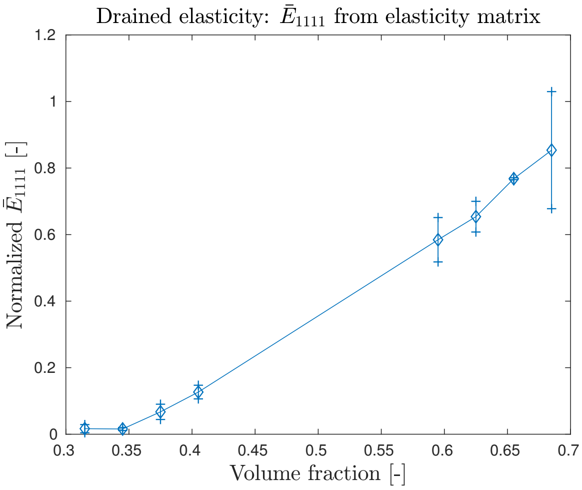


Figure 6.15: $\bar{E}_{1111}$ in drained case. In total 8 statistical ensembles based on 34 microstructure realizations.

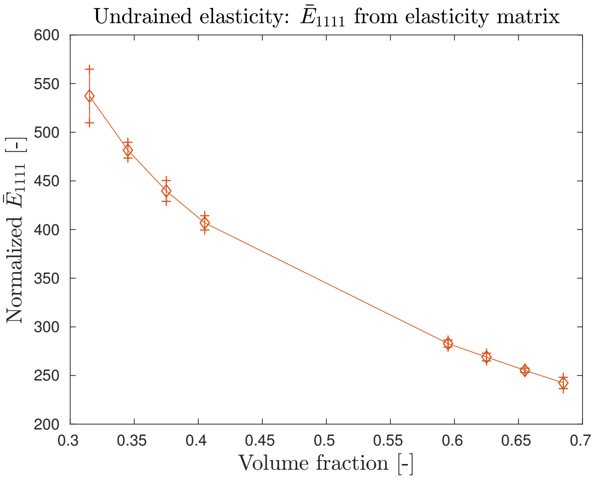


Figure 6.16: $\bar{E}_{1111}$ in undrained case. In total 8 statistical ensembles based on 32 microstructure realizations.

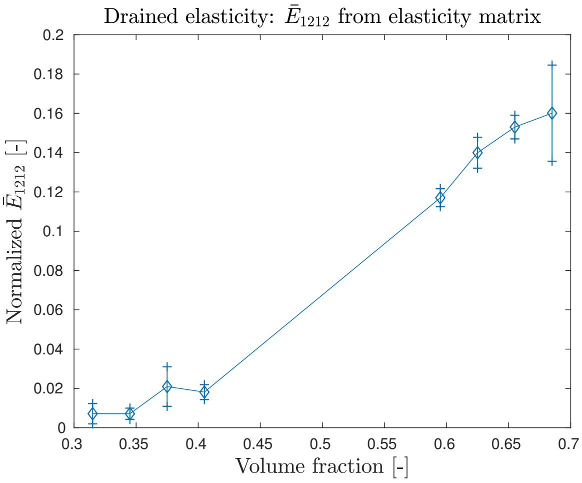


Figure 6.17: $\bar{E}_{1212}$ in drained case. In total 8 statistical ensembles based on 34 microstructure realizations.

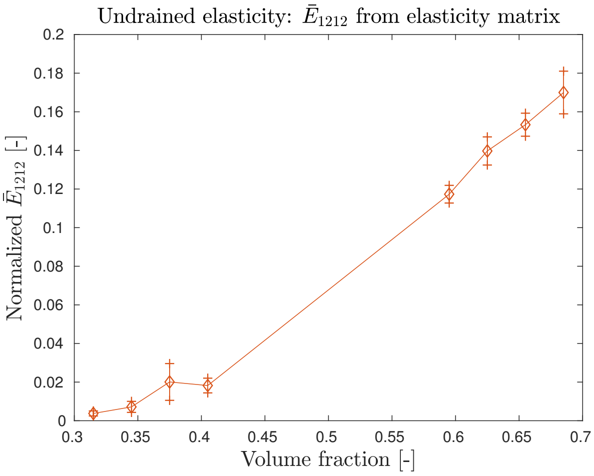


Figure 6.18: $\bar{E}_{1212}$ in undrained case. In total 8 statistical ensembles based on 32 microstructure realizations.

6.2 Effective ionic conductivity of SBE

The isotropic sub-scale constitutive relationship is:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = - \begin{bmatrix} D & 0 & 0 \\ 0 & D & 0 \\ 0 & 0 & D \end{bmatrix} \boldsymbol{\nabla} u \tag{6.5}$$

and the corresponding macro-scale relationship is:

$$\begin{bmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \end{bmatrix} = - \begin{bmatrix} \bar{D}_{11} & \bar{D}_{12} & \bar{D}_{13} \\ \bar{D}_{21} & \bar{D}_{22} & \bar{D}_{23} \\ \bar{D}_{31} & \bar{D}_{32} & \bar{D}_{33} \end{bmatrix} \boldsymbol{\nabla} \bar{u} \tag{6.6}$$

In a similar fashion as for linear elasticity, the effective diffusion coefficient matrix is obtained by setting the macro-scale ion gradient to 1 componentwise. Figure 6.19-6.21 show the simulation results obtained by solving the sub-scale problem on the pore space (porosity 36.94% liquid electrolyte) of the same example structure using the unit ion gradient $\boldsymbol{\nabla}\bar{u} = [1\ 0\ 0]^{\mathrm{T}}$ as the driving force.

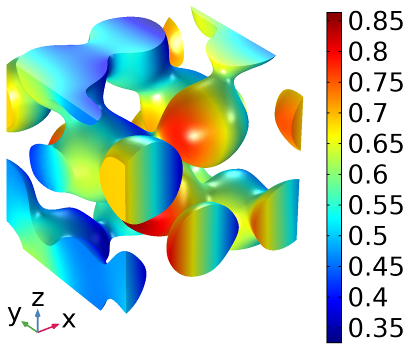


Figure 6.19: Dimensionless fluctuation field u^{S} in liquid electrolyte.

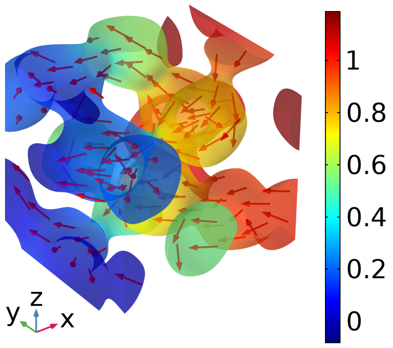


Figure 6.20: Dimensionless ion concentration field u and normalized ion flux vectors $\bar{\boldsymbol{q}}$.

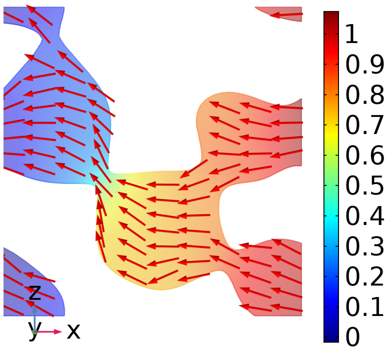


Figure 6.21: Dimensionless ion concentration field u and normalized ion flux projected on the middle xz-plane.

Figure 6.22 shows the relationship between the diffusivity and the porosity. Note that the microstructure porosity ϕ is computed in the following way:

$$\phi = 1 - \mathrm{VF} \tag{6.7}$$

where VF corresponds to the volume fraction of the solid polymer phase.

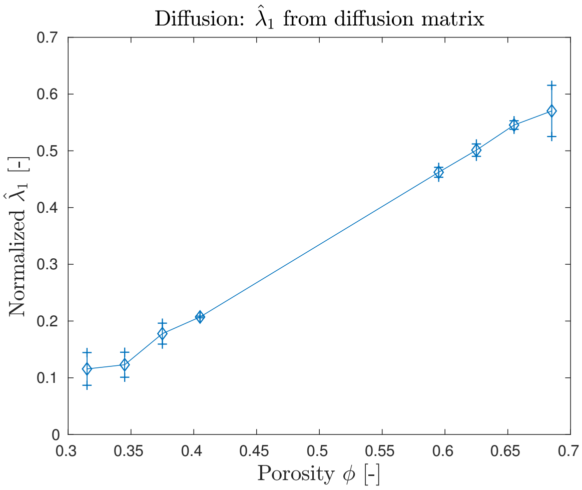


Figure 6.22: Largest normalized eigenvalue $\hat{\lambda}_1$. In total 8 statistical ensembles based on 36 microstructure realizations.

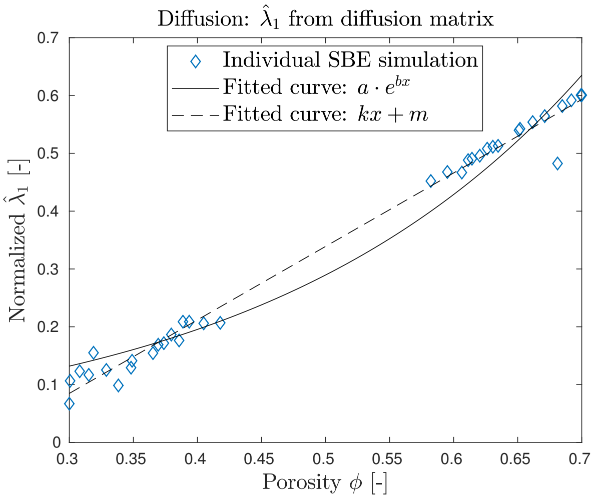


Figure 6.23: Largest normalized eigenvalue $\hat{\lambda}_1$. Individual SBE simulations without VF grouping.

Once again, it can be noted that the SBE diffusivity increases with increasing porosity. Furthermore, it seems that the graph is revealing a linear relationship between the diffusivity and the porosity. See Figure 6.24 and 6.25 to consider the normal component $\bar{D}_{11}$ and off-diagonal component $\bar{D}_{12}$ separately. Note that the effective diffusivity is dominated by the normal components since the off-diagonal components are very small.

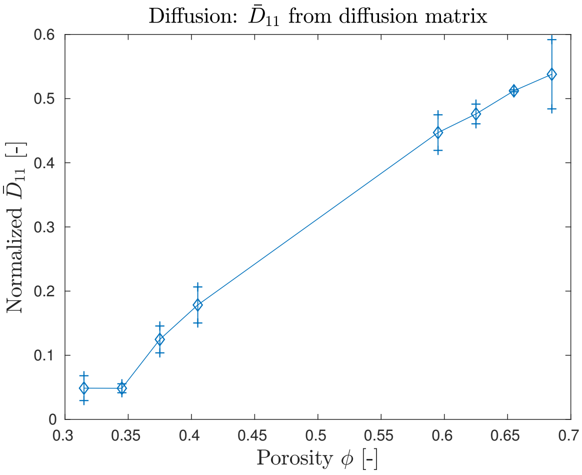


Figure 6.24: Normal component $\bar{D}_{11}$. In total 8 statistical ensembles based on 36 microstructure realizations.

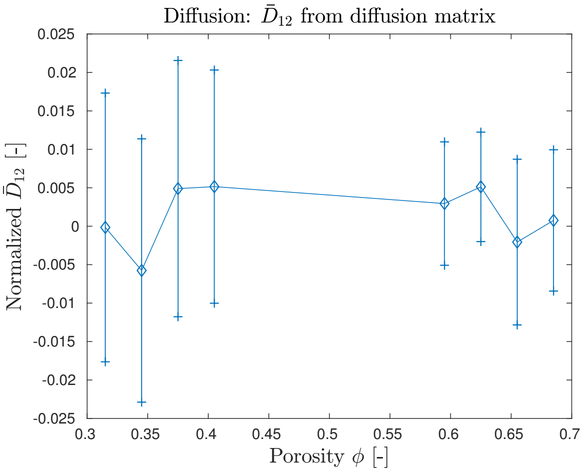


Figure 6.25: Off-diagonal component $\bar{D}_{12}$. In total 8 statistical ensembles based on 36 microstructure realizations.

7 Concluding remarks and future works

In order to evaluate the multifunctional performances of the SBE, a method for generating artificial SBE microstructures was developed. The generated artificial SBE microstructure was utilized as an RVE for computational homogenization. At the end of the project, a semi-automated artificial microstructure generation technique was achieved. Initially 23 unique artificial microstructures were produced, but inverting these resulted in 23 additional structures which in the end adds up to a total of 46 structures. These structures were used in finite element analyses based on linear elasticity and stationary diffusion. In particular WPBC were used in the computational homogenization. Due to meshing and convergence issues, a few of these structures failed. After obtaining the effective properties, the data was visualized in graphs which revealed that the lower bound of the stiffness increases for increasing volume fraction of polymer matrix, while the upper bound of the stiffness and the ionic conductivity decreases. Furthermore, it seems that the ionic conductivity scales linearly with the volume fraction of polymer matrix, while the bounds of the effective stiffness scale non-linearly.

Many obstacles were encountered during the project and several compromises were made. Some issues stem directly from the STL files exported from COMSOL. E.g. structures with many seeds fail more often since COMSOL's STL files tend to become corrupted for complicated structures. Furthermore, structures with many seeds usually result in more small details that are hard to trace and patch. Thus, a compromise of limiting the number of seeds was made. It turns out that 10 seeds correspond to the maximum number of seeds that consistently result in manageable artificial microstructures. A major risk with limiting the number of seeds is that it also limits the amount of statistics that fit in the SVEs. This implies that the effective properties might not be predicted with a reasonable confidence level; the spread might be too large. However, the results show that the spread in general seems quite small. This confirms that 10 seeds actually do result in an SVE with sufficient statistics.

Another issue is the inability to fully automate the microstructure generation process. Some isolevels in COMSOL are completely infeasible to patch since they might contain too small fragments or e.g. zero thickness shell objects that cannot form volumes. Due to the inability to fully automate the microstructure generation, massproducing SVEs for computational homogenization requires too much manual work. This explains why the SVE sample size is limited in this project. However, a brute force strategy can be employed to fully automate the structure generation. The COMSOL STL files can be generated for each increment of the isolevel, then MATLAB can try to patch it up, and lastly COMSOL can try to repair and convert it into a solid. If any error occurs during the process, then the code can simply throw away the current microstructure and move on to the next. One disadvantage with this brute force method is that the fail rate is going to be very high and the majority of the structures will be thrown away, see Algorithm 2 for an overview of the method.

```
Initialize MATLAB
    for n artificial SBE microstructures do
        Start with random placement of seeds and end up with a heat manipulated structure in COMSOL;
        for each discrete increment in isolevel do
            Compute the isosurface and isovolume;
            Export isosurface and isovolume STL files;
            Patch STL files in MATLAB;
            Repair and convert structure in COMSOL;
            Invert structure in COMSOL with boolean operator;
            if any error is encountered along the process then
                Throw away artificial SBE microstructure and move on to the next one;
            else
                Save artificial SBE microstructure
            end
        end
    end
End MATLAB
```

Algorithm 2: Simplified overview of the automated structure generation.

Another major challenge that has yet been addressed is the accuracy of the SBE representation. Currently only two criterions are imposed on the artificial microstructure, i.e. micro-porosity and bicontinuity. However, these characteristics are still far too common and they might apply for other porous materials too. If more knowledge of the SBE can be obtained, then the artificial microstructure can be modified to better represent the real SBE.

The current results are in a general format since they are normalized and dimensionless. This means that as soon as real material parameters are known, it will then be easy to compare the results from this thesis to other experimental studies for validation purposes. Experimental validation is crucial for this type of project since it can be used as a feedback to improve the current artificial SBE microstructure. This can be achieved by e.g. comparing the artificial microstructure to FIB-SEM data, or by performing real material testing on the SBE. Hence, the accuracy of the artificial SBE can be increased as more experimental data is obtained in the future. Throughout this project, the methodology has been to construct an artificial SBE which visually mimics Figure 4.1, but this is a highly subjective method which should be avoided in future works.

In essence, this thesis demonstrates the concept of utilizing the heat equation in order to generate an artificial microstructure. Although the goal of the project is to mimic the SBE in the structural battery, the end result is a quite general recipe on artificial microstructure generation. Any future work that uses this thesis as a starting point should focus on implementing the fully automated structure generation and gathering experimental data to validate against. Nevertheless, the bottom line is that this thesis paves the way for more rigorous artificial SBE microstructure generation in the future.

Bibliography

[1] Fredi et al., "Graphitic microstructure and performance of carbon fibre Li-ion structural battery electrodes", *Multifunctional Materials*, vol. 1, no. 015003, 2018.

[2] Johannisson et al., "Multifunctional performance of a carbon fiber UD lamina electrode for structural batteries", *Composites Science and Technology*, vol. 168, pp. 81-87, 2018.

[3] L. Asp and E. Greenhalgh, "Multifunctional structural battery and supercapacitor composites" in *Multifunctionality of Polymer Composites*, William Andrew, 2015.

[4] N. Ihrner, W. Johannisson, F. Sieland, D. Zenkert and M. Johansson, "Structural Lithium Ion Battery Electrolytes via Reaction Induced Phase-Separation", *Journal of Materials Chemistry A*, vol. 5, no. 48, pp. 25652–25659, 2017.

[5] Chalmers University of Technology, Gothenburg, "Carbon fibre can store energy in the body of a vehicle" 2018. [Online]. Available: https://www.chalmers.se/en/departments/ims/news/Pages/carbon-fibre-can-store-energy.aspx, Accessed on: May 16, 2019.

[6] Snyder et al., "Multifunctional Structural Composite Batteries for U.S. Army Applications" presented at *Proceedings of the 2006 Army Science Conference*, Orlando, Florida, 2006.

[7] T. Carlson, "Multifunctional composite materials: Design, manufacture and experimental characterisation", Ph.D. dissertation, Department of Engineering Sciences and Mathematics, Luleå University of Technology, Luleå, Sweden, 2013.

[8] Asp et al., "A battery half cell, a battery and their manufacture", EP2893581A1 European Patent Office, 2012.

[9] D. Carlstedt, "On the multifunctional performance of structural batteries", Lic. thesis, Industrial and Materials Science, Chalmers University of Technology, Gothenburg, Sweden, 2019.

[10] M. Ekh, R. Jänicke, F. Larsson and K. Runesson, Compendium: "The finite element method - Solid Mechanics", Chalmers University of Technology, Gothenburg, Sweden, 2018.

[11] N. S. Ottosen and H. Petersson, *Introduction to the finite element method*, Wiltshire, United Kingdom: Prentice Hall, 1992.

[12] R. Jänicke, F. Larsson and K. Runesson, Compendium: "Computational Homogenization in Material Mechanics", Chalmers University of Technology, Gothenburg, Sweden, 2019.

[13] F. Larsson, K. Runesson, S. Saroukhani and R. Vafadari, "Computational homogenization based on a weak format of micro-periodicity for RVE-problems", *Computer Methods in Applied Mechanics and Engineering*, vol. 200, pp. 11-26, 2011.

[14] F. Fritzen, T. Böhlke and E. Schnack, "Periodic three-dimensional mesh generation for crystalline aggregates based on Voronoi tessellations.", *Computational Mechanics*, vol. 43, pp. 701-713, 2009.

[15] T. Schüler, R. Jänicke and H. Steeb, "Nonlinear modeling and computational homogenization of asphalt concrete on the basis of XRCT scans.", *Construction and Building Materials*, vol. 109, pp. 96-108, 2016.

[16] L. Spyrou, S. Brisard and K. Danas, "Multiscale modeling of skeletal muscle tissues based on analytical and numerical homogenization.", *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 92, 2018.

[17] C. H. Rycroft, "Voro++: A Three-Dimensional Voronoi Cell Library in C++", *Chaos (Woodbury, N.Y.)*, vol. 19, 2009.